

Apple

\$1.80



Assembly

Line

Volume 7 -- Issue 3

December, 1986

In This Issue...

Eight Ways to Count Bits in a Byte	2
Apple //gs ROM Checksummer	9
Who Worked on the //gs?.	9
Commented Listing of ProDOS, \$DE00-DEF2.	12
Another Update to Bob's Program Selector	20
Beginning to Peek at the //gs Monitor.	21
Corrections to HR to DHR Converter	26
Demo of Two Simple //gs Tools.	27
Woz Re-Codes Hi-Res Address Calculations	29

New ProDOS Book

Dennis Doms and Tom Weishaar, Technical Consultant and Publisher of Open-Apple, have conspired to bring us an interesting new book on programming under ProDOS, especially focussing on BASIC.SYSTEM.

"ProDOS Inside and Out" begins by explaining what an operating system is, progresses by describing files and directories, and goes on into simple commands. The next sections cover Applesoft programming and text file handling, followed by information about using machine language under BASIC.SYSTEM and using the ProDOS Kernel and MLI calls from BASIC.

This book does an excellent job of introducing the basic concepts of ProDOS, and then takes the reader on into quite advanced territory. It's very refreshing to find a book that doesn't assume you're already an expert and still has enough substance to help make you into one.

"ProDOS Inside and Out", by Dennis Doms and Tom Weishaar, from TAB Books. List is \$16.95, we'll have it for \$16 + shipping.

Fight Ways to Count Bits in a Byte.....Bob Sander-Cederlof

Of course, there are always eight bits in a byte, by definition. But sometimes we want to know how many of those eight bits are 1's. There are enough reasons to generate this count that some computers have a special machine language opcode to count the number of 1-bits in a byte or word.

One reason that comes to mind is to compute the odd- or even-parity bit for a byte of ASCII data. Another is in processing of picture data in a computer vision system.

I came up with at least eight different programs to count the 1-bits in a byte. I would choose one based on how critical speed or memory-usage is in a particular case.

The fastest one, table lookup, can translate a byte to a bit count in 4 cycles (7 if you also count getting the byte-value into X or Y. It is fast, but it requires a 256-byte table. The table has the counts for each possible value. If speed is critical, I would use this method. I would probably use one of the other methods to create the table during initialization, rather than assembling it from source code. An example of the table lookup method is shown in lines 2130-2160, as a subroutine.

The next method that I thought of is shown in lines 1500-1590. I count the 1-bits in the X-register, while shifting the data byte. The loop runs eight times, once for each bit position. This one takes from 86 cycles to 94 cycles, depending on how many 1-bits there are. The average time is 90 cycles. (I am not counting the JSR and RTS.)

A slightly faster method checks two bit positions during each loop (lines 1610-1730). The reduction in loop overhead changes the times to minimum 66, maximum 74, average 70 cycles.

The fastest method I found without table lookup is shown in lines 1750-2010. This one is completely unwrapped, so there is no loop overhead at all. The times come out to minimum 43, maximum 51, average 47 cycles.

I tried just optimizing the first method, and lines 2030-2110 are the result. This one gets more average speed because it loops only until there are no more 1-bits left. The minimum is only 8 cycles (when the byte = \$00). If the byte = \$80, it takes 14 cycles. For \$C0 or \$40, it takes 128 or 120 cycles. For values that have bit 0 = 1, it will take from 70 to 77 cycles, depending on the number of 1-bits in the other seven bit positions. The overall average is 65 cycles.

By changing this last method just a little, the overall average can be reduced to between 58 and 59 cycles. The result is still fairly small, so I think this one would be my favorite choice when the blinding speed of a table lookup is not necessary:

S-C Macro Assembler Version 2.0.....DOS \$100, ProDOS \$100, both for \$120
 Version 2.0 DOS Upgrade Kit for 1.0/1.1/1.2 owners.....\$20
 ProDOS Upgrade Kit for Version 2.0 DOS owners.....\$30
 Source Code of S-C Macro 2.0 (DOS only).....additional \$100
 Full Screen Editor for S-C Macro (with complete source code).....\$49
 S-C Cross Reference Utility.....without source code \$20, with source \$50
 RAK-Ware DISASM.....without source code \$30, with source \$50
 S-C Word Processor (with complete source code).....\$50
 DP18 Source and Object.....\$50
 Double Precision Floating Point for Applesoft (with source code).....\$50
 ES-CAPE (Extended S-C Applesoft Program Editor).....
 Including Version 2.0 With Source Code.....\$50
 ES-CAPE Version 2.0 and Source Code Update (for Registered Owners)....\$30
 Bag of Tricks 2 (Quality Software).....(\$49.95) \$45 *
 MacASM -- Macro Assembler for Macintosh (Mainstay).....(\$150.00) \$100 *
 S-C Documentor (complete commented source code of Applesoft ROMs).....\$50
 Cross Assemblers for owners of S-C Macro Assembler.....\$32.50 to \$50 each
 (Available: 6800/1/2, 6301, 6805, 6809, 68000, Z-80, Z-8, 8048,
 8051, 8085, 1802/4/5, PDP-11, G1650/70, others)

AAL Quarterly Disks.....each \$15, or any four for \$45
 Each disk contains the source code from three issues of AAL:
 Jan-Mar, Apr-Jun, Jul-Sep, and Oct-Dec.
 (All source code is formatted for S-C Macro Assembler. Other assemblers
 require some effort to convert file type and edit directives.)

Diskettes (with hub rings)..... package of 25 for \$15 *
 Vinyl disk pages, 6"x8.5", hold two disks each.....10 for \$6 *
 Diskette Mailing Protectors (hold 1 or 2 disks).....40 cents each
 (Cardboard folders designed to fit 6"x9" Envelopes.) or \$25 per 100 *
 Envelopes for Diskette Mailers..... 6 cents each

65802 Microprocessor (Western Design Center).....(\$95) \$50 *
 quickLoader EPROM System (SCRG).....(\$179) \$170 *
 PROMGRAMMER (SCRG).....(\$149.50) \$140 *
 Switch-a-Slot (SCRG).....(\$179.50) \$170 *
 Minuteman 250 Uninterruptible Power Supply.....(\$359) \$320 +

"65816/65802 Assembly Language Programming", Fischer.....(\$19.95) \$18 *
 "Programming the 65816", Eyes.....(\$22.95) \$21 *
 "Apple //e Reference Manual", Apple Computer.....(\$24.95) \$23 *
 "Apple //c Reference Manual", Apple Computer.....(\$24.95) \$23 *
 "ProDOS Technical Reference Manual", Apple Computer.....(\$29.95) \$27 *
 "Apple ProDOS: Advanced Features for Programmers", Little..(\$17.95) \$17 *
 "ProDOS Inside and Out", Doms & Weishaar.....(\$16.95) \$16 *
 "Beneath Apple ProDOS", Worth & Lechner.....(\$19.95) \$18 *
 "Beneath Apple DOS", Worth & Lechner.....(\$19.95) \$18 *
 "Inside the Apple //c", Little.....(\$19.95) \$18 *
 "Inside the Apple //e", Little.....(\$19.95) \$18 *
 "Understanding the Apple //e", Sather.....(\$24.95) \$23 *
 "Understanding the Apple II", Sather.....(\$22.95) \$21 *
 "Apple II+/IIE Troubleshooting & Repair Guide", Brenner.....(\$19.95) \$18 *
 "Apple II Circuit Description", Gayler.....(\$22.95) \$21 *
 "Assem. Language for Applesoft Programmers", Finley & Myers.(\$18.95) \$18 *
 "Enhancing Your Apple II, vol. 1", Lancaster.....(\$15.95) \$15 *
 "Enhancing Your Apple II, vol. 2", Lancaster.....(\$17.95) \$17 *
 "Assembly Cookbook for the Apple II/IIE", Lancaster.....(\$21.95) \$20 *
 "Real Time Programming -- Neglected Topics", Foster.....(\$9.95) \$9 *
 "Microcomputer Graphics", Myers.....(\$14.95) \$14 *
 "Assembly Lines -- the Book", Wagner.....(\$19.95) \$18 *
 "AppleVisions", Bishop & Grossberger.....(\$39.95) \$36 *

* On these items add \$2.00 for the first item and
 \$.75 for each additional item for US shipping.
 + Inquire for shipping cost.
 Foreign customers inquire for postage needed.

Texas residents please add 6 1/8 % sales tax to all orders.

*** S-C SOFTWARE, P. O. BOX 280300, Dallas, TX 75228 ***
 *** (214) 324-2050 ***
 *** Master Card, VISA, Discover and American Express ***

```

COUNT    LDX #0
           LDA BYTE
           BEQ .3      value is 00000000
           BPL .2      bit 7 = 0
.1         INX         count the 1-bit
.2         ASL         shift the value
           BMI .1      ...its 1, count it
           BNE .2      ...its 0, do not count
.3         RTS         finished

```

Lines 2270-2460 implement still another method, which always takes 57 cycles regardless of the mix of zeroes and ones. It uses an extra pagezero location for a temporary cell. The TAX in line 2450 is required only if it is really required that the bit count be returned in the X-register. My test program has this requirement, but there is no reason to force that requirement on a real program. If the A-register is good enough, then this method only takes 55 cycles.

Lines 2180-2250 are "half" a table lookup. That is, I am using half a table: entries for values from \$00 to \$7F. This is a tiny bit slower than the full table lookup, but saves 128 bytes of table. I shift the value right one bit position, guaranteeing a value from \$00 to \$7F, and save the bit shifted out in carry. Then adding carry to the value from the table gives me the count for the whole original value. A slight change to how carry is added can reduce the average time by half a cycle:

```

COUNT    LDA BYTE
           LSR
           TAY
           LDX TBL1,Y
           BCC .1
           INX
.1         RTS

```

The wildest way I came up with is based on one I read about in the latest issue of Byte magazine (the IBM special issue, in the BIX section). By masking, adding, and shifting, the bits can be all aligned and added into a count. Lines 2480-2720 are my code for this method. The time is 58 cycles, regardless of the 0/1 mix.

I wrote a test routine, so I could tell whether my methods really worked or not. Lines 1070-1480 call each of three different bit-counters for every possible value of the byte. It keeps calling the first method until I hit the space bar, and then advances to the second method. Then it keeps calling the second, until another key-tap advances it to the third method. I can keep cycling through the methods this way, until I type a RETURN to end it. Inside each of the three loops I have a "LDA \$C030" instruction, to toggle the speaker. The three loops are identical in timing except for the bit-counting code itself. The result is that I can tell by "ear" which methods are fastest, and which ones have a constant time regardless of the 0/1 mix. I tested the accuracy by comparing the results in TBL1 and TBL2 with the monitor "V" command, and by displaying the TBL using the monitor.

```

1000 #SAVE S.BIT COUNTERS
1010 #-----
00- 1020 BYTE .EQ $00
01- 1030 B .EQ $01
01- 1040 SUM1 .EQ $01
02- 1050 SUM2 .EQ $02
1060 #-----
1070 T
0800- A9 00 1080 LDA #0
0802- 85 00 1090 STA BYTE
1100 #-----
0804- 20 56 08 1110 .1 JSR COUNT.1
0807- A4 00 1120 LDY BYTE
0809- 8A 1130 TXA
080A- 99 00 0A 1140 STA TBL1,Y
080D- AD 30 C0 1150 LDA $C030
0810- E6 00 1160 INC BYTE
0812- AD 00 C0 1170 LDA $C000
0815- 10 ED 1180 BPL .1
0817- 8D 10 C0 1190 STA $C010
081A- C9 8D 1200 CMP #$8D
081C- F0 37 1210 BEQ .99
1220 #-----
081E- 20 64 08 1230 .2 JSR COUNT.2
0821- A4 00 1240 LDY BYTE
0823- 8A 1250 TXA
0824- 99 00 0B 1260 STA TBL2,Y
0827- AD 30 C0 1270 LDA $C030
082A- E6 00 1280 INC BYTE
082C- AD 00 C0 1290 LDA $C000
082F- 10 ED 1300 BPL .2
0831- 8D 10 C0 1310 STA $C010
0834- C9 8D 1320 CMP #$8D
0836- F0 1D 1330 BEQ .99
1340 #-----
0838- 20 76 08 1350 .3 JSR COUNT.3
083B- A4 00 1360 LDY BYTE
083D- 8A 1370 TXA
083E- 99 00 0B 1380 STA TBL2,Y
0841- AD 30 C0 1390 LDA $C030
0844- E6 00 1400 INC BYTE
0846- AD 00 C0 1410 LDA $C000
0849- 10 ED 1420 BPL .3
084B- 8D 10 C0 1430 STA $C010
084E- C9 8D 1440 CMP #$8D
0850- F0 03 1450 BEQ .99
1460 #-----
0852- 4C 04 08 1470 JMP .1
0855- 60 1480 .99 RTS
1490 #-----
1500 COUNT.1
0856- A0 08 1510 LDY #8
0858- A2 00 1520 LDX #0
085A- A5 00 1530 LDA BYTE
085C- 0A 1540 .1 ASL
085D- 90 01 1550 BCC .2
085F- E8 1560 INX
0860- 88 1570 .2 DEY
0861- D0 F9 1580 BNE .1
0863- 60 1590 .1 NEXT BIT
1590 RTS
1600 #-----
1610 COUNT.2
0864- A0 04 1620 LDY #4
0866- A2 00 1630 LDX #0
0868- A5 00 1640 LDA BYTE
086A- 0A 1650 .1 ASL
086B- 10 01 1660 BPL .2
086D- E8 1670 .2 IF BIT = 1, COUNT IT
086E- 90 01 1680 .2 BCC .3
0870- E8 1690 .3 IF BIT = 1, COUNT IT
0871- 0A 1700 .3 INX
0872- 88 1710 .3 ASL
0873- D0 F5 1720 BNE .1
0875- 60 1730 .1 NEXT PAIR OF BITS
1730 RTS

```

Personalize and Protect your Apple with the **Banana Chip**®

When you turn on your Apple™, this remarkable new ROM replacement will display **your** name or personalized message with optional password protection.



\$39.95

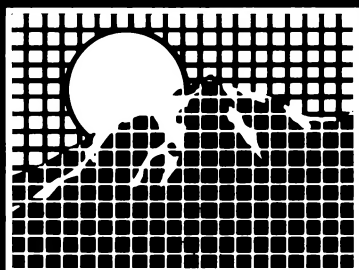
CALL TOLL FREE:

1-800-843-0975

Nationwide

1-800-443-9060

In California



Azure Mountain Products

14755 Ventura Blvd., Ste. 1483 Sherman Oaks, CA 91403

```

1740 #-----
1750 COUNT.3
0876- A2 00 1760 LDX #0 NO LOOPS, JUST STRAIGHT-LINE CODE
0878- A5 00 1770 LDA BYTE
087A- 10 01 1780 BPL .1 BIT 7
087C- E8 1790 INX
087D- 4A 1800 .1 LSR
087E- 90 01 1810 BCC .2 BIT 0
0880- E8 1820 INX
0881- 4A 1830 .2 LSR
0882- 90 01 1840 BCC .3 BIT 1
0884- E8 1850 INX
0885- 4A 1860 .3 LSR
0886- 90 01 1870 BCC .4 BIT 2
0888- E8 1880 INX
0889- 4A 1890 .4 LSR
088A- 90 01 1900 BCC .5 BIT 3
088C- E8 1910 INX
088D- 4A 1920 .5 LSR
088E- 90 01 1930 BCC .6 BIT 4
0890- E8 1940 INX
0891- 4A 1950 .6 LSR
0892- 90 01 1960 BCC .7 BIT 5
0894- E8 1970 INX
0895- 4A 1980 .7 LSR
0896- 90 01 1990 BCC .8 BIT 6
0898- E8 2000 INX
0899- 60 2010 .8 RTS
2020 #-----
2030 COUNT.4
089A- A2 00 2040 LDX #0
089C- A5 00 2050 LDA BYTE
089E- F0 06 2060 BEQ .3
08A0- 10 01 2070 .1 BPL .2
08A2- E8 2080 INX
08A3- 0A 2090 .2 ASL
08A4- D0 FA 2100 BNE .1
08A6- 60 2110 .3 RTS
2120 #-----
2130 COUNT.5
08A7- A4 00 2140 LDY BYTE
08A9- BE 00 0A 2150 LDX TBL1,Y
08AC- 60 2160 RTS
2170 #-----
2180 COUNT.6
08AD- A5 00 2190 LDA BYTE
08AF- 4A 2200 LSR
08B0- AA 2210 TAX
08B1- A9 00 2220 LDA #0
08B3- 7D 00 0A 2230 ADC TBL1,X
08B6- AA 2240 TAX
08B7- 60 2250 RTS
2260 #-----
2270 COUNT.7
08B8- A5 00 2280 LDA BYTE
08BA- 4A 2290 LSR BIT 0 3
08BB- 85 01 2300 STA B 3
08BD- A9 00 2310 LDA #0 3
08BF- 2A 2320 ROL 3
08C0- 46 01 2330 LSR B BIT 1 5
08C2- 69 00 2340 ADC #0
08C4- 46 01 2350 LSR B BIT 2
08C6- 69 00 2360 ADC #0
08C8- 46 01 2370 LSR B BIT 3
08CA- 69 00 2380 ADC #0
08CC- 46 01 2390 LSR B BIT 4
08CE- 69 00 2400 ADC #0
08D0- 46 01 2410 LSR B BIT 5
08D2- 69 00 2420 ADC #0
08D4- 46 01 2430 LSR B BIT 6
08D6- 65 01 2440 ADC B BIT 7
08D8- AA 2450 TAX
08D9- 60 2460 RTS
2470 #-----

```

08DA-	A5	00	2480	COUNT.8		
08DC-	29	55	2490		LDA	BYTE
08DE-	85	01	2500		AND	#\$55
08E0-	45	00	2510		STA	SUM1
08E2-	4A		2520		EOR	BYTE
08E3-	65	01	2530		LSR	
08E5-	85	01	2540		ADC	SUM1
08E7-	29	33	2550		STA	SUM1
08E9-	85	02	2560		AND	#\$33
08EB-	45	01	2570		STA	SUM2
08ED-	4A		2580		EOR	SUM1
08EE-	4A		2590		LSR	
08EF-	65	02	2600		LSR	
08F1-	85	02	2610		ADC	SUM2
08F3-	29	0F	2620		STA	SUM2
08F5-	85	01	2630		AND	#\$0F
08F7-	45	02	2640		STA	SUM1
08F9-	4A		2650		EOR	SUM2
08FA-	4A		2660		LSR	
08FB-	4A		2670		LSR	
08FC-	4A		2680		LSR	
08FD-	65	01	2690		LSR	
08FF-	AA		2700		ADC	SUM1
0900-	60		2710		TAX	
			2720		RTS	
			2730		*-----	
0901-			2740		.BS #+255/256#256-#	
			2750		*-----	
0A00-			2760	TBL1	.BS 256	
0B00-			2770	TBL2	.BS 256	
			2780		*-----	

BITS 6,4,2,0
 BITS 7,5,3,1
 CLEARS CARRY
 FORM wwxxyyzz, where each pair
 is 0, 1, or 2.
 Isolate 00xx00zz
 Isolate ww00yy00
 Now it is 00ww00yy
 Form 0uuu0vvv, where each group
 is 0, 1, 2, 3, or 4.
 Isolate 00000vvv
 Isolate 0uuu0000
 Now it is 00000uuu
 Form count 0-8

RAMWORKS™

**ACCEPT NO SUBSTITUTES.
BECAUSE THERE AREN'T ANY.**

There's only one card like RamWorks. We've got the best hardware design. We supply the best software and we've got the best support from software companies.

If someone tempts you with an imitation, please get both sides of the story. You'll discover why RamWorks offers the best enhancements to AppleWorks and other programs, and at the lowest price.

GUARANTEED!

**214-241-6060
9 AM - 11 PM**

AE
"We Set the Standard"
APPLIED ENGINEERING

Apple //gs ROM Checksummer.....Bob Sander-Cederlof

Somewhere I ran across a statement that the 128K bytes of ROM in the //gs have a standard checksum. A value is stored at \$FFFFFF6 and \$FFFFFF7 to pad out the checksum, so that it will always add up to \$1234. I tried various ways of adding up the bytes to get that value, and came up with the following little program.

```

1000 *SAVE S.CHECKSUM
1010 *-----
1020 .OP 65816
1030 *-----
1040 FORM.AND.DISPLAY.IIGS.CHECKSUM
000800- 18 1050 CLC ENTER NATIVE MODE
000801- FB 1060 XCE
000802- C2 30 1070 REP #$30 FULL 16
1080 *-----
000804- A2 00 00 1090 LDX #0 0 TO FFFF
000807- 8A 1100 TXA START WITH A=0
000808- 38 1110 SEC WHY? IT MAKES ANSWER $1234
000809- 7F 00 00 FE 1120 .1 ADC $FE0000,X BANK $FE
00080D- E8 1130 INX
00080E- E8 1140 INX TWO BYTES AT A TIME
00080F- D0 F8 1150 BNE .1 ...UNTIL WHOLE BANK
000811- 7F 00 00 FF 1160 .2 ADC $FF0000,X BANK $FF
000815- E8 1170 INX
000816- E8 1180 INX TWO BYTES AT A TIME
000817- D0 F8 1190 BNE .2 ...UNTIL WHOLE BANK
000819- 85 00 1200 STA 0 SAVE RESULT AT $00.01
1210 *-----
00081B- 38 1220 SEC EMULATION MODE
00081C- FB 1230 XCE
00081D- A5 01 1240 LDA 1 PRINT CHECKSUM WITH OLD MONITOR
00081F- 20 DA FD 1250 JSR $FD DA SUBROUTINE 'PRBYTE'
000822- A5 00 1260 LDA 0
000824- 20 DA FD 1270 JSR $FD DA
000827- 60 1280 RTS
1290 *-----

```

The 128K ROM occupies the space from \$FE0000 through \$FFFFFF. My program adds up the data there two bytes at a time in 16-bit registers. Doing a normal add of these "words" gave a sum of \$1233, so I started with SEC instead of CLC to get a sum of \$1234. I took my program to a computer store and tried it on a "real" //gs, with a different ROM set, and got the same result: \$1234.

If you make any changes to the ROMs yourself, be sure to fix the checksum too. Otherwise you may not even be able to boot!

Who Worked on the //gs?.....Bob Sander-Cederlof

You can see a list of the names of the people inside Apple who worked on the //gs by typing in the following program. The names are stored inside the //gs ROM, organized according to which project they worked on. I suppose future versions of the //gs ROMs may not have this information (they may need the space for more useful tools), or it may be moved around, but so far it is all the machines I have looked at.

The names start at \$BA0B, and are in two consecutive blocks terminated by a 00 byte. Most of the bytes are ASCII characters with bit 7 high (=1). Whenever you find a byte with bit 7 low (=0), it is a repeat count for the following character. Thus 07 A0 means repeat \$A0 seven times, or print seven blanks. You will also find repeat counts followed by \$53, which is an inverse S. However, in MouseText, it is a horizontal line. Evidently it is supposed to be displayed with MouseText turned on.

There may be a program somewhere inside the ROM that prints the list of names, but I haven't even tried finding it yet. Anyway, the following one will do it.

```

1000 *SAVE S.NAMES.IIGS
1010 *-----
1020 *      PRINT NAMES OF APPLE //GS DEVELOPERS
1030 *-----
      1040      .OP 65816
      1050      *-----
000800- 18      1080      CLC          ENTER NATIVE MODE
000801- FB      1090      XCE
000802- C2 10   1100      REP #$10      X,Y 16-BIT MODE
000804- A9 8D   1101      LDA #$8D      SKIP TWO LINES
000806- 20 3B 08 1102      JSR MY.COUT
000809- 20 3B 08 1103      JSR MY.COUT
00080C- A2 0A BA 1110      LDX ##$BA0A  NAMES START AT $BA0B
00080F- 20 18 08 1120      JSR PRINT.NAMES  FIRST BLOCK
000812- 20 18 08 1130      JSR PRINT.NAMES  SECOND BLOCK
000815- 3B      1140      SEC
000816- FB      1150      XCE          BACK TO NATIVE MODE
000817- 60      1190      RTS
      1200      *-----
      1210      PRINT.NAMES
000818- E8      1220      .1      INX          NEXT CHAR
000819- BF 00 00 FF 1230      LDA $FF0000,X
00081D- F0 07      1240      BEQ .2      ...END OF A BLOCK
00081F- 10 06      1250      BPL .3      ...REPEAT COUNT
000821- 20 3B 08 1260      JSR MY.COUT  PRINT THE CHAR
000824- 80 F2      1270      BRA .1
000826- 60      1280      .2      RTS          RETURN
      1290      *-----
000827- A8      1320      .3      TAY          REPEAT COUNT TO Y-REG
000828- E8      1340      INX
000829- BF 00 00 FF 1350      LDA $FF0000,X  GET REPEATED CHAR
00082D- C9 53      1351      CMP #$53      MOUSE TEXT LINE?
00082F- D0 02      1352      BNE .4      ...NO
000831- A9 AD      1353      LDA #"-"      ...YES, SUBSTITUTE DASH
000833- 20 3B 08 1360      JSR MY.COUT  PRINT THE CHAR
000836- 88      1370      DEY          N TIMES
000837- D0 FA      1380      BNE .4
000839- 80 DD      1390      BRA .1
      1400      *-----
      1410      MY.COUT
00083B- 48      1420      PHA          SAVE EVERYTHING!!!
00083C- DA      1430      PHX
00083D- 5A      1440      PHY
00083E- 08      1450      PHP
00083F- 38      1460      SEC          EMULATION MODE
000840- FB      1470      XCE
000841- 08      1480      PHP
000842- 20 ED FD 1490      JSR $FDED
000845- FB      1500      PLP          ORIGINAL MODE
000846- 28      1510      XCE
000847- 28      1520      PLP          RESTORE REGISTERS
000848- 7A      1530      PLY
000849- FA      1540      PLX
00084A- 68      1550      PLA
00084B- 60      1560      RTS
      1570      *-----

```

7 Meg Iie/1 Meg Iic

Want the best & easiest-to-use (most support) RAM card on the market — at the best price? Buy Checkmate Technology's State-of-the-Art MULTIRAM RGB RAM CARD™ from us w/BATTERY BACKED-UP RAM options that save programs like AppleWorks for 10+ years! FAST CHEAP REPLACEMENT FOR HARD DISKS, USER EXPANDABLE TO 7 MEGABYTES, compatible with all (100%) 3rd party software/hardware, optional 65C816 Co-Processor card (\$147), sharp 80 columns/Double Hi-Res, & FREE RGB! It's a direct substitute for Ramworks I-III™ by Applied Engineering (also known as Preferred Computing) or Apple 80 col cards, has a 5 YR WARRANTY, 15 day money back guarantee, & FREE AUTOMATIC SOFTWARE UPDATES ONLY FROM US!

MultIRam RGB expands to 1 Meg RAM + optional 4 Meg's **MultIRam Plus BATTERY PROTECTED RAM**. Optional 2 Meg's STATIC RAM here soon — for a total of 7 megabytes! **MultIRam Iie** expands to 768k. And the software & manual is the best on the market!

FREE APPLEWORKS EXPANDER SOFTWARE WITH EVERY CARD — loads ALL versions of Appleworks (in 30 seconds w/AW 2.0), even printer routines, runs faster, expands Desktop over 2048k, stores 23,000+ records & 4,800+ word processor lines, auto-segments large files onto multiple disks! **FREE AW TIME/DATE ON-SCREEN** w/ clock, AUTO-COPY TO RAM, PRODOS/DOS 3.3, RAM DISK & RAM TEST. Optional CP/M & Pascal Ram disks (\$20/ea)

	MultIRam RGB Card	MultIRam Iie Card
64k MultIRam	169. ...	129.
320k MultIRam	199. ...	164.
576k MultIRam	229. ...	199.
832k MultIRam	254. ...	234.
1024k MultIRam	279. ← ← ← ←	

	MultIRam Plus Piggyback	
256k MultIRam Plus	220. ...	NEW
512k MultIRam Plus	269. ...	NEW
768k MultIRam Plus	316. ...	NEW
1024k MultIRam Plus	359. ...	NEW
2-4 meg MultIRam Plus	Call	NEW
Battery/AC Kit (for Plus)	36. ...	NEW

Apple Iie Enhancement Kit	62.
Cermetek 1200 Baud internal modem Iie/II+/Iigs	179.
Incomm 2400 Baud external univ. modem	319.
Clockworks Clock (Thunder/Time HO™ compat)	89.
13" RGB Monitor (free cable)	322.
ProAPP 10/20 Meg Hard Disks (+ FREE software)	Call
Apple Compatible Drives Iie/II+ (Iigs/Iic + 15)	135.
NETWORKING UP TO 256 APPLE Iic/Iie/Iigs	Call

Terms: Add \$4-Ground or \$6-Air shipping & phone # to each U.S. MultIRam card order (foreign orders/FPO/APO extra). Add 3% for MasterCard/Visa (include #/expir) & P.O.'s. For fast delivery send Cashier's/Certified check/Money Order. C.O.D. (add \$6) & personal checks accepted (allow 16 days). Tax res add 7 1/4% tax.

Checkmate Technology's **MultIRam CX** card easily expands your Iic to 640k and has a **CX+ Piggyback** to add another 512k (over 1152k total)! It's 100% compatible with all Iic software/hardware & upgradable with a 65C816 kit (\$105) — all using about 50% less power than Z-RAM! It has the **SAME FREE SOFTWARE/UPDATES & 5 YR WARRANTY AS Iie CARDS.**

256k MULTIRAM CX	179.
512k MULTIRAM CX	209.
768k MULTIRAM CX Combo	328.
1024k MULTIRAM CX Combo	354.
256k MULTIRAM CX+	210.
512k MULTIRAM CX+	222.

AutoWorks (Best AppleWorks Macro+) ...	37.
1200 Baud Iic Modem (+\$20/cable)	159.
Clip-on Iic Rechargeable Battery	124.
Iic Flat Panel Display (C-Vue)	299.
Iic System Clock	66.

WHY BUY FROM COIT VALLEY COMPUTERS RATHER THAN SOME MAIL ORDER HOUSES? Only we offer a **15 day MultIRam complete money back satisfaction guarantee**, extra software, more support, **free automatic software updates**, free 64k w/ each 256k/512k/768k Iie/RGB card. We know the products, & stock them for fast shipping! **CALL FOR QUESTIONS, PRICES, DISCOUNTS. SCHOOLS & GROUPS WELCOME.**

Autoworks, Ramworks III/Timemaster II H.O./Z-ram, respective trademarks of Software Touch, Applied Engineering. *Call for RGB Connectors/cables (not included).

ORDER NOW!

COIT VALLEY COMPUTERS (214) 234-5047
14055 Waterfall Way Dallas, Texas 75240

NAME _____		
ADDRESS _____		
CITY _____	STATE _____	ZIP _____
PHONE () _____		
SIGNATURE _____		
QTY	DESCRIPTION	PRICE
MC/VS _____	SHIP. _____	
EXP _____	TOTAL _____	

COIT VALLEY COMPUTERS
(214) 234-5047

MCI 2969684
Telex 6502969684 (WUI)

14055 Waterfall Way
Dallas, Texas 75240

What happens when you call ProDOS MLI? In assembly language, MLI calls look like this:

```
JSR $BF00
.DA #command,IOB.Address
```

The instruction at \$BF00 is a "JMP \$BFB7" in ProDOS 1.1.1; it is possibly different in other versions. All of the following disassembly is for ProDOS 1.1.1. The changes in the new ProDOS 1.2 are minor, and if you have 1.2 you should be able to figure out what they are.

At \$BFB7 there is some code I call LC-BRIDGE.ENTRY. It "remembers" what language card areas are switched in at \$D000 and at \$E000, and then turns on the language card so that it can jump into the MLI call processor.

```
BFB7: SEC      Set flag
      ROR MLI.ACTIVE.FLAG
      LDA $E000
      STA E000.BYTE (BFF4)
      LDA $D000
      STA D000.BYTE (BFF5)
      LDA $C08B
      LDA $C08B
      JMP $DE00
```

Now comes the good part. The following listing is of the code starting at \$DE00, which decodes the bytes following your JSR \$BF00 and performs your request.

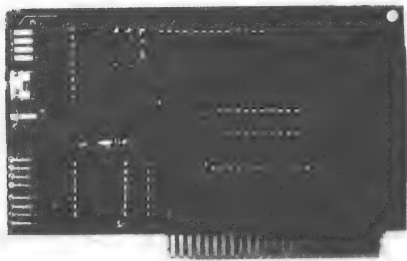
Lines 1010-1080 define some page-zero variables used by MLI.
Lines 1090-1220 define some items in the system global page.
Lines 1230-1280 define some entry points inside the rest of MLI, not listed here.

MLI calls don't change the X and Y registers, so they are saved at line 1390. The return address (of the JSR \$BF00) is pulled off the stack and saved at PARM.PNTR in page zero, so that it can be used to access your command code and IOB address. Lines 1410-1490 also compute the address of the next instruction, to be used later for a return address. This address is saved in the system global page, and is useful sometimes for debugging. (We have published several articles on enhanced error messages and tracers for MLI calls in previous issues of AAL.)

Lines 1500-1650 convert the command code to an index by a strange scheme. The legal command codes are (in hex): 40, 41, 65, 80 thru 82, and C0 thru D3. The hashing algorithm used here adds the high nybble of the command code to the whole code, and then masks it to the lower five bits. This compresses the range of the codes, without any overlapping.

40,41 --> 04,05	C0-CF --> 0C-1B
65 --> 0B	D0-D2 --> 1D-1F
80-82 --> 08-0A	D3 --> 00

The first clock card to add time and date to AppleWorks™ screens and files.



Timemaster H.O. So many features its become the world's best selling clock card.

It's the smart way to put the time and date on your Apple IIe* or II+*. Because only the Timemaster H.O. packs *ALL* the important features of all the competition *combined*, including leap year, year (not just in PRO-DOS), month, date, day of week, hours, minutes, seconds and milliseconds. It's totally PRO-DOS, DOS 3.3, PASCAL and CP/M compatible. And, of course, it works better than any other clock with AppleWorks.

With the Timemaster H.O., you're assured of reliable, trouble-free operation along with simple, one-command time setting, accuracy through leap years, and a firmware maintained interrupt setting.

If you're using or writing software for other clock cards, you're still covered. Because the H.O. will *automatically* emulate them. And only the Timemaster H.O. adds 15 new commands to BASIC. The H.O. even comes complete with two disks full of sample programs, including a computerized appointment book, a DOS dating program, interrupt programs, and over 30 programs that others charge you extra for — or don't even offer. As a low-cost option, you can add true BSR remote control to the H.O., giving you remote control of up to 16 lights and appliances in your home or office.

- Time in hours, minutes, seconds and milliseconds; date with year, month, day of week and leap year • 24-hour military format or 12 hour AM/PM format • Eight software controlled interrupts so you can run two programs at the same time (many examples included) • Can be set to the second • Allows AppleWorks to time and date stamp all data automatically • Displays time and date on AppleWorks screens and files • Five-year warranty

Timemaster H.O. \$129.00
BSR option (may be added later) \$ 39.00

Call to order today, 9 a.m. to 11 p.m. seven days, or send check or money order to Applied Engineering, MasterCard, VISA and C.O.D. welcome. Texas residents add 5% sales tax. Add \$10.00 outside U.S.A.

AE Applied Engineering
The Apple enhancement experts.

P.O. Box 798, Carrollton, TX 75006 (214) 241-6060

Viewmaster 80

Now run AppleWorks™ on your II+ *and* keep full 80 column compatibility!



	Basic Software	Multi- Disk Support	Low Power Mode	80 Column Mode	1 + 1 Soft Sector	Local Disk II	80 Column Emulation	Remote Character
VIEWMASTER 80	-	-	-	-	-	-	-	-
SUPRTERM	-	-	-	-	-	-	-	-
WIZARD 80	-	-	-	-	-	-	-	-
VISION 80	-	-	-	-	-	-	-	-
OMNIVISION	-	-	-	-	-	-	-	-
VIEWMAX 80	-	-	-	-	-	-	-	-
SMARTERM	-	-	-	-	-	-	-	-
VIDEOTERM	-	-	-	-	-	-	-	-

One look at the chart will give you some of the reasons there's only one smart choice in 80 column cards for your Apple. But the real secret to Viewmaster 80's success is something even better: Total compatibility.

Each Viewmaster 80 includes our powerful AppleWorks expand software, allowing AppleWorks to run on the II+ with only 64K (or more) memory. (We recommend the RamFactor memory card, but any compatible Apple memory card will work.) The software provides our full range of AppleWorks enhancements, including expanded records, word processor, multi-disk saving, time and date display on screen with any PRO-DOS clock, and more!

The Viewmaster 80 works with all 80 column applications, including DOS 3.3, PRO-DOS, CP/M, Pascal, WordStar, Format II, Applewriter II, dBase II, Visicalc, Multiplan, and *hundreds* of others — including AppleWorks.

Here are just a few of the powerful features the Viewmaster 80 delivers for a great price (\$139):

- 80 Characters by 24 lines • Fully compatible with all Apple languages and software • Highest compatibility with existing 80 column software • Very low power consumption • High speed (18 MHz) scroll rate • Upper and lower case characters and true descenders, both inverse and normal; all on-screen editing functions are supported • User-definable cursor shape • Compatible with Apple II, II+ and IIe • Five-year warranty

Call today to order or for more information, 9 a.m. to 11 p.m. seven days, or send check or money order to Applied Engineering, MasterCard, VISA and C.O.D. welcome. Texas residents add 5% sales tax. Add \$10.00 outside U.S.A.

AE Applied Engineering
P.O. Box 798, Carrollton, TX 75006
(214) 241-6060

This index is used then to look into the COMMAND.HASH.TABLE, which has the actual command codes in the indexed positions. If the original code is not found there, then the original code was an illegal command number. The hash index is also used to look up the parameter count in PARM.CNT.TABLE. I have appended the code for these two tables to the end of today's listing, at lines 3100 to the end.

Lines 1810-1920 branch various ways according to the command code. Most of the commands are not shown in this listing, but most of the code for READ BLOCK and WRITE BLOCK is shown (lines 2690-3080). When a command is finished, it eventually finds its way back to EXIT.TO.CALLER at line 2180.

Lines 2180-2560 get us back to our own code again, after the JSR \$BF00. If the MLI call produced an error, the code number for that error will be in SYS.ERRNUM. The error code will be returned in the A-register, with carry SET. If there is no error to report, A=0 and carry is clear.

We will probably be presenting more sections of MLI disassembly in the near future. You may remember that we published portions of an earlier ProDOS version back in November and December of 1983.

```

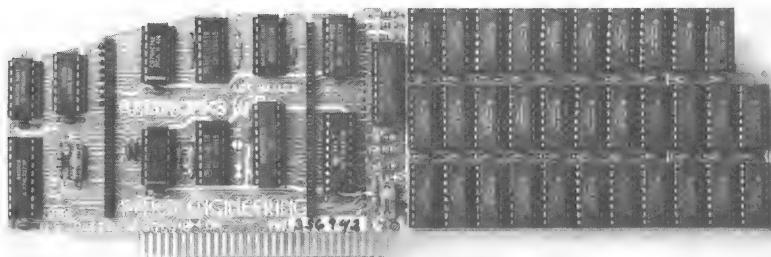
1000 *SAVE S.MLI.DE00.DEF2
1010 *-----
40- 1020 PARM.PNTR .EQ $40.41
42- 1030 COMMAND .EQ $42
43- 1040 UNIT.NO .EQ $43
44- 1050 BUFF.PNTR .EQ $44.45
46- 1060 BLOCK.NO .EQ $46.47
48- 1070 GEN.PNTR1 .EQ $48.49
4E- 1080 GEN.PNTR2 .EQ $4E.4F
1090 *-----
BF03- 1100 CALL.QUIT .EQ $BF03
BF06- 1110 CALL.TIME .EQ $BF06
BF09- 1120 CALL.SYSERR .EQ $BF09
BF0F- 1130 SYS.ERRNUM .EQ $BF0F
BF10- 1140 DRIVER.ADDR.TABLE .EQ $BF10 thru BF2F
BF95- 1150 BACKUP.BIT .EQ $BF95
BF9B- 1160 MLI.ACTIVE.FLAG .EQ $BF9B
BF9C- 1170 MLI.RETURN .EQ $BF9C.D
BF9E- 1180 MLI.X .EQ $BF9E
BF9F- 1190 MLI.Y .EQ $BF9F
BFA0- 1200 LC.BRIDGE.EXIT .EQ $BFA0
BFF4- 1210 E000.BYTE .EQ $BFF4
BFF5- 1220 D000.BYTE .EQ $BFF5
1230 *-----
DEF3- 1240 INTERRUPT.HANDLER .EQ $DEF3
E047- 1250 FILING.FUNCTIONS .EQ $E047
FC9F- 1260 CHECK.IF.MEM.FREE .EQ $FC9F
1270 *-----
FEF5- 1280 JUMP .EQ $FEF5.6
1290 *-----
1300 .OR $DE00
1310 .TA $800
1320 *-----
1330 * JSR $BF00 comes here
1340 * .DA #$xx command byte
1350 * .DA xxxx IOB Address
1360 *-----
1370 MLI.ENTRY
DE00- D8 1380 CLD
DE01- 8C 9F BF 1390 STY MLI.Y
DE04- 8E 9E BF 1400 STX MLI.X
DE07- 68 1410 PLA GET RETURN ADDRESS
DE08- 85 40 1420 STA PARM.PNTR WILL POINT AT BYTES
DE0A- 18 1430 CLC FOLLOWING JSR $BF00
DE0B- 69 04 1440 ADC #4 COMPUTE ACTUAL RETURN
DE0D- 8D 9C BF 1450 STA MLI.RETURN AND SAVE FOR LATER

```

DE10-	68		1460	PLA	
DE11-	85	41	1470	STA	PARM.PNTR+1
DE13-	69	00	1480	ADC	#0
DE15-	8D	9D BF	1490	STA	MLI.RETURN+1
			1500	*---Check Command Code-----	
DE18-	A0	00	1510	LDY	#0
DE1A-	8C	0F BF	1520	STY	SYS.ERRNUM
DE1D-	C8		1530	INY	
DE1E-	B1	40	1540	LDA	(PARM.PNTR).Y
DE20-	4A		1550	LSR	Hash it (CC/16 + CC) & \$1F
DE21-	4A		1560	LSR	
DE22-	4A		1570	LSR	
DE23-	4A		1580	LSR	
DE24-	18		1590	CLC	
DE25-	71	40	1600	ADC	(PARM.PNTR).Y
DE27-	29	1F	1610	AND	#\$1F
DE29-	AA		1620	TAX	Use hashcode as index
DE2A-	B1	40	1630	LDA	(PARM.PNTR).Y Original command code
DE2C-	DD	65 FD	1640	CMP	COMMAND.HASH.TABLE,X
DE2F-	D0	76	1650	BNE	ERR.CALL.NO Not valid command
			1660	*---Get IOB Address-----	
DE31-	C8		1670	INY	
DE32-	B1	40	1680	LDA	(PARM.PNTR).Y
DE34-	48		1690	PHA	
DE35-	C8		1700	INY	
DE36-	B1	40	1710	LDA	(PARM.PNTR).Y
DE38-	85	41	1720	STA	PARM.PNTR+1
DE3A-	68		1730	PLA	
DE3B-	85	40	1740	STA	PARM.PNTR
			1750	*---Check Parm Count-----	
DE3D-	A0	00	1760	LDY	#0
DE3F-	BD	85 FD	1770	LDA	PARM.CNT.TABLE,X
DE42-	F0	1C	1780	BEQ	MLI.GETTIME ...only one with 0 parms
DE44-	D1	40	1790	CMP	(PARM.PNTR).Y
DE46-	D0	63	1800	BNE	ERR.PARM.CNT
			1810	*---Branch Various Ways-----	
DE48-	BD	65 FD	1820	LDA	COMMAND.HASH.TABLE,X
DE4B-	C9	65	1830	CMP	#\$65
DE4D-	F0	0E	1840	BEQ	.1 ...QUIT CALL
DE4F-	0A		1850	ASL	
DE50-	10	14	1860	BPL	MLI.RWBLK \$80 or \$81
DE52-	B0	1D	1870	BCS	MLI.CX.AND.DX \$Cx or \$Dx
DE54-	4A		1880	LSR	\$40 or \$41
DE55-	29	03	1890	AND	#\$03
DE57-	20	F3 DE	1900	JSR	INTERRUPT.HANDLER
DE5A-	4C	78 DE	1910	JMP	EXIT.TO.CALLER
DE5D-	4C	03 BF	1920	JMP	CALL.QUIT \$65
			1930	*-----	
			1940	* Command \$82. Get the Date and Time	
			1950	*-----	
			1960	MLI.GETTIME	
DE60-	20	06 BF	1970	JSR	CALL.TIME
DE63-	4C	78 DE	1980	JMP	EXIT.TO.CALLER
			1990	*-----	
			2000	* Commands \$80 and \$81	
			2010	*-----	
			2020	MLI.RWBLK	
DE66-	4A		2030	LSR	Make \$00 and 01
DE67-	69	01	2040	ADC	#1 Into \$01 and 02
DE69-	85	42	2050	STA	COMMAND Store into command block
DE6B-	20	B2 DE	2060	JSR	BLOCK.IO.SETUP Do the I/O
DE6E-	4C	78 DE	2070	JMP	EXIT.TO.CALLER
			2080	*-----	
			2090	* Commands \$C0 thru \$D3	
			2100	*-----	
			2110	MLI.CX.AND.DX	
DE71-	4A		2120	LSR	Make command code into
DE72-	29	1F	2130	AND	#\$1F an index
DE74-	AA		2140	TAX	
DE75-	20	47 EO	2150	JSR	FILING.FUNCTIONS
			2160	*---fall into EXIT routine-----	
			2170	* (DE78) DE5A DE63 DE6E DEB0 callers	
			2180	EXIT.TO.CALLER	
DE78-	A9	00	2190	LDA	#0 Clear BACKUP bit
DE7A-	8D	95 BF	2200	STA	BACKUP.BIT
DE7D-	AC	0F BF	2210	LDY	SYS.ERRNUM If any error code,
DE80-	C0	01	2220	CPY	#1 then set carry
DE82-	98		2230	TYA	and clear Z-bit
DE83-	08		2240	PHP	Save this status

RamWorks® III

Patented Performance from the Recognized Leader



With battery backed RAM port, RGB port, increased memory capacity, full software compatibility and more compact design, RamWorks III is a generation ahead.

RamWorks III is the newest 3rd generation RAM card for the Apple IIe. It incorporates all of the technology and improvements that years of experience and over a hundred thousand sales have given us. By selling more memory cards than anyone else and listening to our customers, we were able to design a memory card that has the ultimate in performance, quality, compatibility and ease of use. A design so advanced it's patented. We call it RamWorks III, you'll call it awesome!

The AppleWorks Amplifier.

While RamWorks III is recognized by all memory intensive programs, NO other expansion card comes close to offering the multitude of enhancements to AppleWorks that RamWorks III does. Naturally, you'd expect RamWorks III to expand the available desktop, after all Applied Engineering was a year ahead of everyone else *including Apple* in offering more than 55K, and we still provide the largest AppleWorks desktops available. But a larger desktop is just part of the story. Look at all the AppleWorks enhancements that even Apple's own card does not provide and *only* RamWorks III does. With a 256K or larger RamWorks III, *all* of AppleWorks (including printer routines) will automatically load itself into RAM dramatically increasing speed by eliminating the time required to access the program disk drive. Switch from word processing to spreadsheet to database at the speed of light with no wear on disk drives.

Only RamWorks eliminates AppleWorks' internal memory limits, increasing the maximum number of records available from 1,350 to over 25,000. *Only* RamWorks increases the number of lines permitted in the word processing mode from 2,250 to over 15,000. And *only* RamWorks offers a built-in printer buffer, so you won't have to wait for your printer to stop before returning to AppleWorks. RamWorks even expands the clipboard. And auto segments large files so they can be saved on two or more disks. You can even have Pinpoint or MacroWorks and your favorite spelling checker in RAM for instant response.

RamWorks, *nothing* comes close to enhancing AppleWorks so much.

The Most Friendly, Most Compatible Card Available.

Using RamWorks III couldn't be easier because it's compatible with more off-the-shelf software than any other RAM card. Popular programs like AppleWorks, Pinpoint, Catalyst, MouseDesk, Howard-Sofi, FlashCalc, Pro-Filer, Managing Your Money, SuperCalc 3a, and MagiCalc to name a few (and *all* hardware add on's like ProFile and Sider hard disks). RamWorks is even compatible with software written for Apple cards. But unlike other cards, RamWorks plugs into the IIe auxiliary slot providing our super sharp 80 column text (U.S. Patent #4601081) in a completely integrated system while leaving expansion slots 1 through 7 available for other peripheral cards.

RamWorks III is compatible with all

Apple IIe's, enhanced, unenhanced, American or European versions.

Highest Memory Expansion.

Applied Engineering has always offered the largest memory for the IIe and RamWorks III continues that tradition by expanding to 1 full MEG on the main card using standard RAMs, more than most will ever need (1 meg is about 500 pages of text)...but if you do ever need more than 1 MEG, RamWorks III has the widest selection of expander cards available. Additional 512K, 2 MEG, or 16 MEG cards just snap directly onto RamWorks III by plugging into the industry's only low profile (no slot 1 interference) fully decoded memory expansion connector. You can also choose non-volatile, power independent expanders allowing permanent storage for up to 20 years.

It Even Corrects Mistakes.

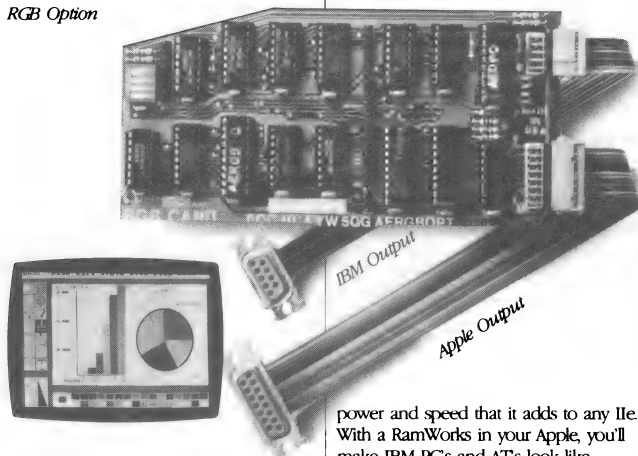
If you've got some other RAM card that's not being recognized by your programs, and you want RamWorks III, you're in luck. Because all you have to do is plug the memory chips from your current card into the expansion sockets on RamWorks to recapture most of your investment!

The Ultimate in RGB Color.

RGB color is an option on RamWorks and with good reason. Some others combine RGB color output with their memory cards, but that's unfair for those who don't need RGB *and* for those that do. Because if you don't need RGB

Applied Engineering doesn't make you buy it, but if you want RGB output you're in for a nice surprise because the RamWorks RGB option offers better color graphics plus a more readable 80 column text (that blows away any composite color monitor). For only \$129 it can be added to RamWorks giving you a razor sharp, vivid brilliance that most claim is the best they have ever seen. You'll also appreciate the multiple text colors (others only have green) that come standard. But the RamWorks RGB option is more than just the ultimate in color output because unlike others, it's fully compatible with all the Apple standards for RGB output control, making it more compatible with off-the-shelf software. With its FCC certified design, you can use almost any RGB monitor because only the new RamWorks RGB option provides both Apple standard and IBM standard RGB outputs (cables included). The RGB option plugs into the back of RamWorks with no slot 1 inter-

RGB Option



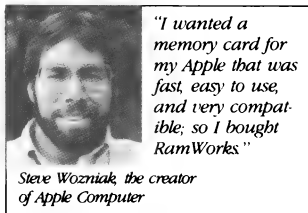
ference and remember you can order the RGB option with your RamWorks or add it on at a later date.

True 65C816 16 Bit Power.

RamWorks III has a built-in 65C816 CPU port for direct connection to our optional 65C816 card. The only one capable of linearly addressing more than 1 meg of memory for power applications like running the Lotus 1-2-3™ compatible program, VIP Professional. Our 65C816 card does not use another slot but replaces the 65C02 yet maintains full 8 bit compatibility.

Endorsed by the Experts.

A+ magazine said "Applied Engineering's RamWorks is a boon to those who must use large files with AppleWorks...I like the product so much that I am buying one for my own system." inCider magazine said "RamWorks is the most



powerful auxiliary slot memory card available for your IIe, and I rate it four stars...For my money, Applied Engineering's RamWorks is king of the hill."

Apple experts everywhere are impressed by RamWorks's expandability, versatility, ease of use, and the sheer

power and speed that it adds to any IIe. With a RamWorks in your Apple, you'll make IBM PCs and AT's look like slowpokes.

Quality and Support of the Industry Leader

RamWorks III is from Applied Engineering, the largest, most well supported manufacturer of Apple peripherals and the inventor of large RAM cards for the Apple. With our 5 year no hassle warranty and outstanding technical support, you're assured of the most trouble free product you can buy.

It's Got It All

- 15 day money back guarantee
- 5 year hassle free warranty insures

coverage no matter where you purchase

- Built-in super sharp 80 column display, (U.S. Patent #4601081)
- Expandable to 1 MEG on main card
- Expandable to 16 meg with expander cards, with NO slot 1 interference
- Can use 64K or 256K RAMs
- Powerful linear addressing 16 bit coprocessor port
- Automatic AppleWorks expansion up to 3017K desktop
- Accelerates AppleWorks
- Built-in AppleWorks printer buffer
- The only large RAM card that's 100% compatible with all IIe software
- RamDrive™ the ultimate disk emulation software included free
- Memory is easily partitioned allowing many programs in memory at once
- Compatible, RGB option featuring ultra high resolution color graphics and multiple text colors, with cables for both Apple and IBM type monitors
- Built-in self diagnostics software
- Lowest power consumption (U.S. Patent #4601081)
- Takes only one slot (auxiliary) even when fully expanded
- Socketed and user upgradeable
- Software industry standard
- Advanced Computer Aided Design
- Used by Apple Computer, Steve Wozniak and virtually all software companies
- Displays date and time on the AppleWorks screen with any PRO-DOS compatible clock
- Much, much more!

RamWorks III with 64K	\$179
RamWorks III with 256K	\$219
RamWorks III with 512K	\$269
RamWorks III with 1 MEG	\$369
RamWorks III with 1.5 MEG	\$539
RamWorks III with 2 to 16 MEG	CALL
65C816 16 Bit Card	\$159
RGB Option	\$129
Optional Software:	
Pinpoint with RAM	
Enhancement Software	\$79
VIP Professional	\$219

RamWorks III. The industry standard for memory expansion of the Apple IIe. ORDER YOUR RamWorks III TODAY. 9 a.m. to 11 p.m. 7 days, or send check or money order to Applied Engineering, MasterCard, Visa and C.O.D. welcome. Texas residents add 5½% sales tax. Add \$10.00 if outside U.S.A.

AE Applied Engineering™
P.O. Box 798, Carrollton, TX 75006
(214) 241-6060

```

DE84- 78      2250      SEI          Disable IRQ's
DE85- 4E 9B BF 2260      LSR MLI.ACTIVE.FLAG Clear this flag
DE88- 68      2270      PLA          Get saved status
DE89- AA      2280      TAX          and keep it in X-reg
DE8A- AD 9D BF 2290      LDA MLI.RETURN+1
DE8D- 48      2300      PHA          Put return address on stack
DE8E- AD 9C BF 2310      LDA MLI.RETURN
DE91- 48      2320      PHA
DE92- 8A      2330      TXA          Now push the status for RTI
DE93- 48      2340      PHA
DE94- 98      2350      TYA          Get error code in A-reg
DE95- AE 9E BF 2360      LDX MLI.X      Restore X and Y
DE98- AC 9F BF 2370      LDY MLI.Y
DE9B- 48      2380      PHA          Error code on stack
DE9C- AD F4 BF 2390      LDA E000.BYTE
DE9F- 4C A0 BF 2400      JMP LC.BRIDGE.EXIT
2410      *-----*
2420      * LC.BRIDGE.EXIT is code at $BFA0 in
2430      * the system global page. It restores
2440      * the language card to the state it
2450      * was in when JSR $BF00 was executed.
2460      *-----*
2470      * LC.BRIDGE.EXIT EOR $E000
2480      * BEQ .1 BFAA
2490      * STA $C082
2500      * BNE .2 BFB5
2510      * .1 LDA D000.BYTE $BFF5
2520      * EOR $D000
2530      * BEQ .2 BFB5
2540      * LDA $C083
2550      * .2 PLA
2560      * RTI
2570      *-----*
2580      ERR.NO.DEVICE
DEA2- A9 28      2590      LDA #$28      "NO DEVICE CONNECTED"
DEA4- 20 09 BF 2600      JSR CALL.SYSERR
2610      ERR.CALL.NO
DEA7- A9 01      2620      LDA #1      "BAD CALL TYPE"
DEA9- D0 02      2630      BNE DEAD
2640      ERR.PARM.CNT
DEAB- A9 04      2650      LDA #4      "BAD PARAMETER COUNT"
DEAD- 20 D7 DE 2660      DEAD JSR CALL.CALL.SYSERR
DEB0- B0 C6      2670      BCS EXIT.TO.CALLER ... ALWAYS
2680      *-----*
2690      BLOCK.IO.SETUP
DEB2- A0 05      2700      LDY #5      COPY REST OF COMMAND BLOCK
DEB4- 08      2710      PHP          FROM IOB TO ZERO-PAGE
DEB5- 78      2720      SEI          DO NOT ALLOW IRQ'S
DEB6- B1 40      2730      .1 LDA (PARM.PNTR),Y
DEB8- 99 42 00 2740      STA COMMAND.Y
DEBB- 88      2750      DEY
DEBC- D0 F8      2760      BNE .1
DEBE- A6 45      2770      LDX BUFF.PNTR+1
DECO- 86 4F      2780      STX GEN.PNTR2+1
DEC2- E8      2790      INX
DEC3- E8      2800      INX
DEC4- A5 44      2810      LDA BUFF.PNTR
DEC6- F0 01      2820      BEQ .2
DEC8- E8      2830      INX
DEC9- 20 9F FC 2840      .2 JSR CHECK.IF.MEM.FREE
DECC- B0 08      2850      BCS .3      ...NOT FREE
DECE- 20 DA DE 2860      JSR BLOCK.IO
DED1- B0 03      2870      BCS .3      ...I/O ERROR
DED3- 28      2880      PLP          RESTORE IRQ STATUS
DED4- 18      2890      CLC          NO ERRORS
DED5- 60      2900      RTS
2910      *-----*
2920      .3 PLP          RESTORE IRQ STATUS
DED6- 28      2930      CALL.CALL.SYSERR JSR CALL.SYSERR
DED7- 20 09 BF 2940      *-----*
2950      * (DEDA) DECE ECOA EE83 FOE4 F475 callers
2960      BLOCK.IO
DEDA- A5 43      2970      LDA UNIT.NO Clean this up a little
DEDC- 29 F0      2980      AND #$F0
DEDE- 85 43      2990      STA UNIT.NO
DEEO- 4A      3000      LSR          Make it into index too
DEE1- 4A      3010      LSR
DEE2- 4A      3020      LSR
DEE3- AA      3030      TAX

```

```

DEE4- BD 10 BF 3040 LDA DRIVER, ADDR. TABLE, X
DEE7- 8D F5 FE 3050 STA JUMP
DEEA- BD 11 BF 3060 LDA DRIVER, ADDR. TABLE+1, X
DEED- 8D F6 FE 3070 STA JUMP+1
DEFO- 6C F5 FE 3080 JMP (JUMP)
3090 *-----
3100 .OR $FD65
3110 .TA $800
3120 COMMAND.HASH.TABLE

FD65- D3 00 00
FD68- 00 40 41
FD6B- 00 00
FD6D- 80 81 82
FD70- 65 C0 C1
FD73- C2 C3
FD75- C4 C5 C6
FD78- C7 C8 C9
FD7B- CA CB
FD7D- CC CD CE
FD80- CF 00 D0
FD83- D1 D2

FD85- 02 FF FF
FD88- FF 02 01
FD8B- FF FF
FD8D- 03 03 00
FD90- 04 07 01
FD93- 02 07
FD95- 0A 02 01
FD98- 01 03 03
FD9B- 04 04
FD9D- 01 01 02
FDA0- 02 FF 02
FDA3- 02 02

3130 .HS D3.00.00.00.40.41.00.00
3140 .HS 80.81.82.65.C0.C1.C2.C3
3150 .HS C4.C5.C6.C7.C8.C9.CA.CB
3160 .HS CC.CD.CE.CF.00.D0.D1.D2
3170 PARM.CNT.TABLE
3180 .HS 02.FF.FF.FF.02.01.FF.FF
3190 .HS 03.03.00.04.07.01.02.07
3200 .HS 0A.02.01.01.03.03.04.04
3210 .HS 01.01.02.02.FF.02.02.02
3220 *-----

```

We Make Measurement And Control Easy!

12 BIT, 16 CHANNEL, PROGRAMMABLE GAIN A/D

- All new 1984 design incorporates the latest in state-of-art I.C. technologies.
- Complete 12 bit A/D converter, with an accuracy of 0.02%!
- 16 single ended channels (single ended means that your signals are measured against the Apple's GND) or 8 differential channels. Most all the signals you will measure are single ended.
- 9 software programmable full scale ranges, any of the 16 channels can have any range at any time. Under program control, you can select any of the following ranges: ± 10 volts, $\pm 5V$, $\pm 2.5V$, $\pm 1.0V$, $\pm 500mV$, $\pm 250mV$, $\pm 100mV$, $\pm 50mV$, or $\pm 25mV$.
- Very fast conversion (25 micro seconds).
- Analog input resistance greater than 1,000,000 ohms.
- Laser-trimmed scaling resistors.
- Low power consumption through the use of CMOS devices.
- The user connector has +12 and -12 volts on it so you can power your sensors.
- Only elementary programming is required to use the A/D.
- The entire system is on one standard size plug in card that fits neatly inside the Apple.
- System includes sample programs on disk.

PRICE \$319

A few applications may include the monitoring of ● flow ● temperature ● humidity ● wind speed ● wind direction ● light intensity ● pressure ● RPM ● soil moisture and many more.

A/D & D/A

A/D & D/A Features:

- Single PC card
- 8 channels A/D
- 8 channels D/A
- Superfast conversion time
- Very easy programming
- Many analog ranges
- Manual contains sample applications

A/D SPECIFICATIONS

- 0.1% accuracy
- On-board memory
- Fast conversion (0.78 MS per channel)
- A/D process totally transparent to Apple (looks like memory)
- User programmable input ranges are 0 to 10 volts, 0 to 5, -5 to +5, -2.5 to +2.5, -5 to 0, -10 to 0.

The A/D process takes place on a continuous, channel sequencing basis. Data is automatically transferred to its proper location in the on-board RAM. No A/D converter could be easier to use.

D/A SPECIFICATIONS

- 0.1% accuracy
- On-board memory
- On-board output buffer amps can drive 5 mA
- D/A process is totally transparent to the Apple (just poke the data)
- Fast conversion (0.03 MS per channel)
- User programmable output ranges are 0 to 5 volts and 0 to 10 volts.

The D/A section contains 8 digital to analog converters, with output buffer amplifiers and all interface logic on a single card. On-card latches are provided for each of the eight D/A converters. No D/A converter could be easier to use. The on-board amplifiers are laser-trimmed during manufacture, thereby eliminating any requirement for on-set nulling.

PRICE \$199

SIGNAL CONDITIONER

Our 8 channel signal conditioner is designed for use with both our A/D converters. This board incorporates 8 F.E.T. op-amps, which allow almost any gain or offset. For example, an input signal that varies from 2.00 to 2.15 volts or a signal that varies from 0 to 50 mV can easily be converted to 0-10V output for the A/D.

The signal conditioner's outputs are on a high quality 16 pin gold I.C. socket that matches the one on the A/D's so a simple ribbon cable connects the two. The signal conditioner can be powered by your Apple or from an external supply.

FEATURES

- 4.5" square for standard card cage and 4 mounting holes for standard mounting. The signal conditioner does not plug into the Apple, it can be located up to 1/2 mile away from the A/D.
- 22 pin 156-spacing edge card input connector (extra connectors are easily available i.e. Radio Shack)
- Large bread board area.
- Full detailed schematic included.

PRICE \$79

I/O 32

- Provides 4, 8-Bit programmable I/O Ports
- Any of the 4 ports can be programmed as an input or an output port
- All I/O lines are TTL (0-5 volt) compatible
- Your inputs can be anything from high speed logic to simple switches
- Programming is made very easy by powerful on-board firmware
- The I/O 32 is your best choice for any control application

The I/O manual includes many programs for inputs and outputs.

Some applications include:

Burglar alarm, direction sensing, use with relays to turn on lights, sound buzzers, start motors, control tape recorders and printers, use with digital joystick.

PRICE \$89

Please see our other full page ad in this magazine for information on Applied Engineering's Time-Master Clock Card and other products for the Apple.

Our boards are far superior to most of the consumer electronics made today. All I.C.'s are in high quality sockets with mil-spec. components used throughout. P.C. boards are glass-epoxy with gold contacts. Made in America to be the best in the world. All products compatible with Apple II and IIe.

Applied Engineering's products are fully tested with complete documentation and available for immediate delivery. All products are guaranteed with a no hassle three-year warranty.

Texas Residents Add 5% Sales Tax
Add \$10.00 if Outside U.S.A.

Send check or money order to
APPLIED ENGINEERING
P.O. Box 798
Carrollton, TX 75006

Call (214) 241-6060
9 a.m. to 11 p.m. 7 days a week
MasterCard, Visa & C.O.D. Welcome
No extra charge for credit cards

The following refers back to the new ProDOS Quit Code I wrote and published in the July 86 issue of AAL. It has been very popular, judging from the number of letters and phone calls we have received.

Eric Trehus (T'n'T Software) pointed out that I ignored one or more of the conventions Apple established for Quit-Code Program Selectors. On page 87 of the ProDOS Technical Reference Manual, the paragraph with number 2 states that the name of the system program should be stored in a buffer at \$280, starting with a length byte. The first paragraph on page 88 says any non-standard Quit Code must begin with a CLD instruction, so programs can tell who loaded them.

If you want the CLD instruction there, go ahead and insert one between lines 1310 and 1320. I have not found it necessary for any programs I use.

Eric says that when going from BASIC.SYSTEM to APLWORKS.SYSTEM he needed the program name stroed in \$280. I have never run into the problem, but it is easy to fix. Eric suggested inserting the following two lines:

```
2065     STA $280
2125     STA $280,X
```

[Eric's change takes six bytes, so you need to be sure the code still fits in \$300 bytes.]

If you do it Eric's way, only the name of the system file gets stored, without any prefix. I wondered whether or not a full pathname should be there, so I consulted Gary Little's "Apple ProDOS--Advanced Features" book. On page 141, near the bottom, he says either a full or a partial pathname should be put at \$280. We can get the full pathname into \$280 without Eric's two lines, by simply changing line 4860 from "PATHNAME .BS 64" to "PATHNAME .EQ \$280". This is my preference.

When I was trying out the above, I stumbled across a problem. If my Selector finds no SYS or DIR files in a directory, it still displays the pathname and prompt messages. If you then type the RETURN key, it may try to execute garbage, or try various other things. The only valid keystroke when no files are listed is ESCAPE, which will take you back to the list of volume names. Adding two lines makes it go there without displaying the empty list:

```
1771     TXA         see if any files listed
1772     BEQ .2       ...none listed, start over
```

We noticed the other day that when we ran Erv Edge's correction to my program (Aug 86, page 1), we reversed the information. We said change line 3390 from BNE .1 to BPL .1; actually, it is the other way around: change from BPL to BNE. Most of you figured that out already, but we are sorry for the confusion.

Beginning to Peek at the //gs Monitor.....Bob Sander-Cederlof

The //gs Monitor has a lot of new features not found in any earlier model Apple. Unfortunately, you do not get any documentation about the monitor with your machine! Next year you will be able to buy a book that will tell you about it, but who wants to wait?

If you go into the monitor and try some of the old commands, you will find that most of them work. Memory display now shows an additional two digits of address, the bank number, and then a slash and the rest of the address. You can enter addresses the same way, so you can display any memory in the entire 16-megabyte range. For example, to disassemble ROM inside the monitor at \$FDF0, type "FF/FDF0L". To look at the range of memory in bank FE from 0 to FF, type "FE/0.FF" and a <RETURN>. Note that the disassembler output looks a little different now. There are no dollar signs for hex values, and all opcodes for the 65816 are disassembled. Also note that in memory range display, you get both hex and ASCII values for each byte. If you are in 80-column mode, range display shows 16 bytes per line rather than only 8.

The new monitor preserves almost all the standard entry points, so they are clues to deciphering the rest. Looking at the routines TOSUB (FF/FFBE) and NXTITM (FF/FF73) I found the new addresses for the command character and branch tables. The command characters are in coded form at FF/F98B and following, and the branches are at FF/F9AE and following. There are 35 commands now, a fact learned by the disassembly of NXTITM.

I wrote a program to decipher the contents of these two tables and print the results. It takes a little work, because the characters in the table are not in ASCII. They correspond to ASCII values exclusive-ORed with \$B0 and diminished by \$89, which takes place inside the GETNUM subroutine (FF/FFA7). The addresses are the low-bytes only of entry points in page \$FE of bank \$FF (FF/FExx). These addresses must be incremented by one to get the real entry points, because TOSUB uses them by pushing them on the stack and doing an RTS. Anyway, the following program does all the unraveling for you, and prints 35 lines of commands in the order they appear in the tables, showing the entry points for each.

Lines 1110-1170 are an overall loop which runs 35 times, to print the 35 commands. The rest is a subroutine to print one command. By removing the stars from lines 1210-1230, you can get the output in two columns. An advantage to this is that it all fits on one screen.

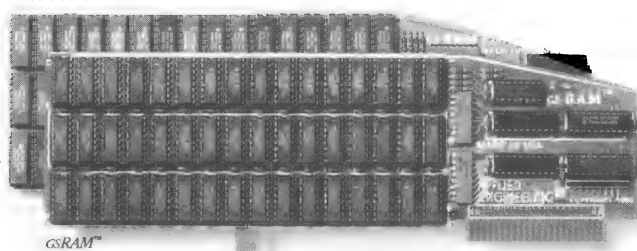
```

1000 *Save s.mon.cmd.tbl
1010 *-----
FDDED- 1020 COUT .EQ $FDED
FDDEA- 1030 PRBYTE .EQ $FDDEA
FDDEE- 1040 CROUT .EQ $FDDEE
1050 *-----
F98B- 1060 LTRS .EQ $F98B      Encoded Table of Letters
F9AE- 1070 ADR.LO .EQ $F9AE    Command starts at $FExx+1
1080 *-----
1090 .OP 65816
1100 *-----
1110 PRINT.MONITOR.COMMAND.TABLE
000800- AO 00 1120 LDY #0
000802- 20 0D 08 1130 .1 JSR PRINT.ONE.COMMAND
```

Insist on **gsRAM™** When You Buy Your **IIgs™**

*Expand the IIgs RAM and ROM with the **gsRAM** or **gsRAM Plus**
with ROM Pak. Available now with 256K to 8 MEG!*

gsRAM Plus™



Remember the 16K cards for the II+ and the 64K cards for the IIe? At the time, that much memory seemed like a lot. But when the owners of these memory cards came to us for more memory, many had to throw away their smaller Apple memory cards or try to sell them. Most of our customers told us that had they known about Applied Engineering's larger memory cards when they bought their Apple, they would have purchased them at the same time.

gsRAM and gsRAM Plus are available now, allowing up to 8 MEG of memory expansion. That's 8 times the memory capacity of Apple's card and just look at the benefits that only gsRAM and gsRAM Plus have over Apple's card:

- Lower cost
- Has 6 RAM banks (Apple's card has 4)
- Has memory expansion port
- Has ROM expansion port
- No configuration blocks to set
- No soldered in RAM chips
- Expandable to 8 MEG
- Expands AppleWorks internal limits
- Built-in Hi-Res self-diagnostic software
- 5 year hassle free warranty (Apple has a 90 day warranty)
- Made in USA

gsRAM for More AppleWorks Power

Only gsRAM and gsRAM Plus eliminates AppleWorks internal memory limits, increasing the maximum number of records available from 6,000 to over 25,000 and only gsRAM and gsRAM Plus increases the number of lines permitted in the word processing mode from 6,000 to over 15,000. And only gsRAM and gsRAM Plus offers a built-in printer buffer so you can continue using AppleWorks while your printer is printing. gsRAM and gsRAM Plus even expand the number of lines in the clipboard from 255 to 2047 and will auto segment large files so they can be saved on two or more disks. You can

even have Pinpoint or MacroWorks and your favorite spelling checker in RAM for instant response. gsRAM and gsRAM Plus will even display the time and date right on the AppleWorks screen. Nothing comes close to enhancing AppleWorks so much.

Turn Your IIgs into a Giant

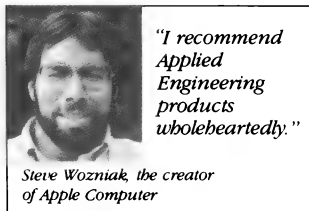
Simply plug gsRAM into the IIgs memory expansion slot and you've got up to 8 megabytes of RAM at your fingertips—all of it instantly and automatically recognized by the IIgs. gsRAM is compatible with all IIgs systems, including AppleWorks, as well as BASIC®, ProDOS, DOS 3.3, PASCAL®, "C" and CP/M®.

Grow by Bytes or Megabytes

We offer gsRAM in two configurations so you can increase your memory 256K at a time (gsRAM) or a megabyte at a time (gsRAM Plus). Both offer full compatibility, lower cost than other boards, and easy expandability. And both are extremely low in power consumption. A fully expanded gsRAM operates at only 375 ma, and gsRAM Plus at only 270 ma (even with 6 megabytes on board!)

gsRAM—for Normal Memory Requirements

gsRAM is available with 256K, 512K, 1 MEG or 1.5 MEG of memory already on board. If you don't need the full 1.5 MEG now, you can choose a gsRAM with less memory and expand it up to 1.5 MEG in the future—or upgrade to gsRAM Plus for a small charge.



*"I recommend
Applied
Engineering
products
wholeheartedly."*

*Steve Wozniak, the creator
of Apple Computer*

With an optional piggyback card, you can expand gsRAM even higher than 1.5 MEG! (Other cards are only expandable to 1 MEG.)

gsRAM Plus—for Growing by Leaps and Bounds

gsRAM Plus is the first Apple memory card to use 1 MEG RAM chips on the main board. It's available with 1 to 6 MEG on board. If you don't need the whole 6 MEG now, you can buy a gsRAM Plus with less memory and easily expand it in the future.

gsRAM Plus can be expanded up to 8 MEG with an optional piggyback card.

Easy Expansion

Both gsRAM and gsRAM Plus use standard RAM chips that are readily available and just plug right in. So unlike other cards, you'll find expanding your gsRAM or gsRAM Plus easy, convenient and very economical. And with our optional ROM expansion module you can even increase the IIgs's ROM space and all in just one slot.

Quality and Support of the Industry Leader

Applied Engineering is the largest supplier of Apple peripherals in the world. We invented the first large RAM cards for the Apple. With a 5-year "no-hassle" warranty and outstanding technical support, you can be sure gsRAM and gsRAM Plus will deliver the performance you're looking for—or return them within 15 days for a full refund.

gsRAM with 256K	\$169
gsRAM with 512K	\$219
gsRAM with 1 MEG	\$299
gsRAM with 1.5 MEG	\$379
gsRAM with 2-8 MEG	CALL
gsRAM Plus with 1-8 MEG	CALL

Order today!

See your dealer or call Applied Engineering today, 9 a.m. to 11 p.m. 7 days. Or send check or money order to Applied Engineering, MasterCard, VISA and C.O.D. welcome. Texas residents add 5% sales tax. Add \$10.00 outside USA.

AE APPLIED ENGINEERING™
The Apple enhancement experts.
(214) 241-6060

P.O. Box 798, Carrollton, TX 75006

CP/M is a registered trademark of Digital Research, Inc.

```

000805- C8      1140      INY
000806- C0 23    1150      CPY #35      There are 35 commands in table
000808- 90 F8    1160      BCC .1
00080A- 4C 8E FD 1170      JMP CROUT
1180      #-----
1190      PRINT.ONE.COMMAND
1200      #---REMOVE ***** FOR 2 COLUMNS-----
1210      *** TYA      CHECK IF ODD OR EVEN
1220      *** LSR
1230      *** BCS .0
1240      #-----
00080D- 20 8E FD 1250      JSR CROUT
1260      #---TAB 10 SPACES-----
000810- A2 0A    1270      LDX #10      TAB OVER 10 SPACES
000812- A9 A0    1280      LDA #" "
000814- 20 ED FD 1290      .1 JSR COUT
000817- CA      1300      DEX
000818- 10 FA    1310      BPL .1
1320      #---Convert char to ASCII-----
00081A- B9 8B F9 1330      LDA LTRS,Y      Value from table
00081D- 38      1340      SEC
00081E- E9 89    1350      SBC #$89      Reverse process from GETNUM
000820- 49 B0    1360      EOR #$B0
1370      #---Prepare char to print-----
000822- A2 A0    1380      LDX #" "      Space before regular chars
000824- C9 A0    1390      CMP #$A0
000826- B0 04    1400      BCS .2      ...not control-char
000828- A2 DE    1410      LDX #"^"      Caret before control-chars
00082A- 09 40    1420      ORA #$40      Make control-char printable
1430      #---Print the char-----
00082C- 48      1440      .2 PHA      Save char itself
00082D- 8A      1450      TXA      Print Space or Caret
00082E- 20 ED FD 1460      JSR COUT
000831- 68      1470      PLA      Print char
000832- 20 ED FD 1480      JSR COUT
1490      #---Print the address-----
000835- A2 03    1500      LDX #3      PRINT " $FE"
000837- BD 47 08 1510      .3 LDA FE,X
00083A- 20 ED FD 1520      JSR COUT
00083D- CA      1530      DEX
00083E- 10 F7    1540      BPL .3
000840- B9 AE F9 1550      LDA ADR.LO,Y
000843- 1A      1560      INC      Add 1 because it needs it
000844- 4C DA FD 1570      JMP PRBYTE
1580      #-----
000847- C5 C6 A4 A0 1590 FE .AS -/EF$ /

```

Seeing all the commands is nice, but it would be easier to read the list if they were in alphabetical order. I modified the program a little, sorted them, and printed them in the order of their ASCII values. Lines 1110-1250 now have two loops. The first one goes through the 35 commands, and stores them into two "sorting trays". I first emptied one of the "trays", by storing zeroes in all 256 locations. Then my SORT.ONE.COMMAND subroutine stores the command ASCII code into the "tray" at the position indexed by the ASCII value itself. The address byte goes into the other "tray" at the same position.

When all 35 commands have been placed into the appropriate positions in the two trays, I run another loop to print out all the non-empty positions. There it is! Simple as can be, they are sorted almost instantaneously.

Then I tried to sort them using the same technique but in the order of the addresses. This did not work, because some addresses are used by more than one command. Only the last command using a particular address printed out. Sigh....

```

1000 *Save s.moncmds.sort
1010 #-----
1020 COUT .EQ $FDED
1030 PRBYTE .EQ $FD8E
1040 CROUT .EQ $FD8E

```

```

1050 *-----
F98B- 1060 LTRS .EQ $F98B      Encoded Table of Letters
F9AE- 1070 ADR.LO .EQ $F9AE    Command starts at $FExx+1
1080 *-----
      1090 .OP 65816
      1100 *-----
000800- 20 1D 08 1110 PRINT.IIGS.MONITOR.COMMANDS.SORTED
000803- A0 00 1120 JSR CLEAR.SORTING.TRAY.A
000805- 20 26 08 1130 LDY #0
000808- C8 1140 .1 JSR SORT.ONE.COMMAND
000809- C0 23 1150 INY
00080B- 90 F8 1160 CPY #35      There are 35 commands in table
      1170 BCC .1
      1180 *---Print the commands-----
00080D- A0 00 1190 LDY #0
00080F- B9 72 08 1200 .2 LDA SORTING.TRAY.A,Y
000812- F0 03 1210 BEQ .3
000814- 20 3A 08 1220 JSR PRINT.ONE.COMMAND
000817- C8 1230 .3 INY
000818- D0 F5 1240 BNE .2
00081A- 4C 8E FD 1250 JMP CROUT
      1260 *-----
      1270 CLEAR.SORTING.TRAY.A
      1280 LDX #0
00081F- 9E 72 08 1290 .1 STZ SORTING.TRAY.A,X
000822- E8 1300 INX
000823- D0 FA 1310 BNE .1
000825- 60 1320 RTS
      1330 *-----
      1340 SORT.ONE.COMMAND
      1350 *---Convert char to ASCII-----
000826- B9 8B F9 1360 LDA LTRS,Y      Value from table
000829- 38 1370 SEC
00082A- E9 89 1380 SBC #$89      Reverse process from GETNUM
00082C- 49 B0 1390 EOR #$B0
00082E- AA 1400 TAX
00082F- 9D 72 08 1410 STA SORTING.TRAY.A,X      It is the sorting index
      1420 *-----
000832- B9 AE F9 1430 LDA ADR.LO,Y
000835- 1A 1440 INC
000836- 9D 72 09 1450 STA SORTING.TRAY.B,X
000839- 60 1460 RTS
      1470 *-----
      1480 PRINT.ONE.COMMAND
      1490 JSR CROUT
00083A- 20 8E FD 1500 *---TAB 10 SPACES-----
00083D- A2 0A 1510 LDX #10      TAB OVER 10 SPACES
00083F- A9 A0 1520 LDA #" "
000841- 20 ED FD 1530 .1 JSR COUT
000844- CA 1540 DEX
000845- 10 FA 1550 BPL .1
      1560 *---Convert char to ASCII-----
000847- B9 72 08 1570 LDA SORTING.TRAY.A,Y
      1580 *---Prepare char to print-----
00084A- A2 A0 1590 LDX #" "      Space before regular chars
00084C- C9 A0 1600 CMP #$A0
00084E- B0 04 1610 BCS .2
000850- A2 DE 1620 LDX #"^"      ...not control-char
000852- 09 40 1630 ORA #$40      Caret before control-chars
      1640 *---Print the char-----
000854- 48 1650 .2 PHA      Save char itself
000855- 8A 1660 TXA      Print Space or Caret
000856- 20 ED FD 1670 JSR COUT
000859- 68 1680 PLA      Print char
00085A- 20 ED FD 1690 JSR COUT
      1700 *---Print the address-----
00085D- A2 03 1710 LDX #3      PRINT " $FE"
00085F- BD 6E 08 1720 .3 LDA FE,X
000862- 20 ED FD 1730 JSR COUT
000865- CA 1740 DEX
000866- 10 F7 1750 BPL .3
000868- B9 72 09 1760 LDA SORTING.TRAY.B,Y
00086B- 4C DA FD 1770 JMP PRBYTE
      1780 *-----
00086E- C5 C6 A4 A0 1790 FE .AS -/EF$ /
      1800 *-----
000872- 1810 SORTING.TRAY.A .BS 256
000972- 1820 SORTING.TRAY.B .BS 256

```


To boldly go at speeds no man has gone before.

*With the TransWarp™ accelerator, your IIe™ or II+™
is 40% faster than the IIGs!*

Computing at warp speed!

It's an experience you shouldn't miss. And with TransWarp, you won't have to. Because TransWarp will run your IIe or II+ software *3.6 times faster* — that's 40% faster than Apple's IIGs!

No more yawning while your Apple™ slowly rearranges text or calculates spreadsheets. With 256K of ultra-fast RAM, TransWarp speeds up *all* Apple software — including AppleWorks, and all educational software, graphics and games. And it's compatible with all standard peripheral cards (such as RamWorks and Apple memory cards), hard disks, 3½" UniDisk, 80-column cards, modems, clock cards, mouses and more! You name it, TransWarp accelerates it. There's even a 16 bit upgrade chip available.



"TransWarp is great! I have replaced all my other accelerators with it!"

*Steve Wozniak, the creator
of Apple Computer*

An important difference.

With TransWarp, software runs up to three times faster than with other cards, since the others can't accelerate programs in auxiliary memory. That's why TransWarp is so much faster than the rest. Nearly all of today's more powerful programs run partially or completely in auxiliary memory: programs like AppleWorks, Pinpoint, Managing Your Money, SuperCalc 3a, BPI and Pascal, to name a few. Why settle for a card that accelerates only part of the memory?

There's one more important difference. Since TransWarp doesn't use memory caching, you get *consistent* high speed performance.

A cinch to use.

Simply plug TransWarp into any slot in your Apple II, II+ or IIe — including slot 3 in the IIe. Instantly you'll be computing

at speeds you only dreamed about before. And should you ever wish to run at normal speed, simply press the ESC key while turning your Apple on.

Since TransWarp is completely transparent, you won't need preboot disks or special software. It's ready to go right out of the package!

Speed = Productivity

Imagine the productivity gains you'll achieve when your programs are running over three times faster. TransWarp is so powerful, your Apple will make IBM PCs™ and even ATs™ look like slowpokes.

- 3.6 MHZ 65C02
- 256K of ultra-fast on-board RAM
- Accelerates main and auxiliary memory
- Low power consumption for cool operation
- Totally transparent operation with all software
- Plugs into any slot, including slot 3 on the Apple IIe
- Accelerated 16 bit option available
- 5-year warranty

Satisfaction guaranteed!

Give your Apple the TransWarp advantage. With our risk-free 15-day money back guarantee, you have nothing to lose but wasted time. Call today!

TransWarp accelerator	\$279
16 bit upgrade (may add later)	\$89

For fast response:

Call Applied Engineering, 9 a.m. to 11 p.m., 7 days at (214) 241-6060. MasterCard, VISA and C.O.D. welcome. Texas residents add 5½% sales tax. Add \$10.00 if outside U.S.A. Or mail check or money order to Applied Engineering, P.O. Box 798, Carrollton, TX 75006.

AE Applied Engineering
The Apple enhancement experts.

P.O. Box 798, Carrollton, TX 75006 (214) 241-6060

Corrections to HR to DHR Converter

David Johnson just called to point out a couple of corrections to his program that converts hi-res graphic images to double hi-res. First, we somehow managed to lose a line of his code. You need to add this line:

```
2435          LSR XLATE.MONO.AUX,X
```

Another spot to change is line 2120. The LDA #\$03 should really be a LDA #\$02, since 3 specifies a full color image and 2 specifies black and white. David reports that 3 has always worked OK for him, so this may not make a real difference, but the specification (ProDOS Technical Note #13) calls for a 2.

An error also crept into the text. Near the end of the fourth paragraph the article says that the AUXMEM portion of the picture is copied into main memory at \$4000-5FFF. What the program really does is copy main memory from \$2000-3FFF to \$4000-5FFF, and then transfer the AUXMEM segment into main memory at \$2000-3FFF.

EnterSoft

Basic-like macros for the S-C Macro Assembler.

Available for the Apple II, II+, IIe, and IIC.

Uses whatever DOS your S-C Macro Assembler uses.

Soon to be released for PRODOS.

Source code included with every disk!!!

Cost? A paltry \$30.00!!!

To Order - Send Check or Money Order to:



Mark Manning
c/o Baggy Games
P. O. Box 591894
Houston, Texas 77259-1894

Apple is a registered Trademark of Apple Computer Inc.

The S-C Macro Assembler is produced by The S-C Software Corporation.

EnterSoft is produced by Mark Manning/Simulacron I/Baggy Games

Demo of Two Simple //gs Tools.....Bob Sander-Cederlof

The //gs Toolbox is chock full of useful tools, and no doubt you have heard about a lot of them by now. However, all of the books and articles I have found so far just describe the tools, without showing any actual addresses or tool numbers so they can be called. Most frustrating!

If you remember the article last month about reading and writing the battery RAM in the //gs, you will remember the general way all tools are called. You must be in full-16 native mode, set up the stack in just the right way, put a tool code in the X-register, and do a JSL \$E10000.

The tool code is made up of two bytes. The low-order byte is the tool set number, and the high-order byte is the tool number in that tool set. One of the missing items of information in most documentation I have seen so far is the list of tool set numbers. As near as I can figure it all out at this time, the following numbers seem to be established:

Set #	Tool
-----	-----
\$01	Tool Locator itself
\$02	Memory Manager
\$03	Miscellaneous Tools
\$04	Graphics Core (QuickDraw?)
\$05	Desk Manager
\$06	Event Manager
\$07	Scheduler
\$08	Sound Manager (Ensoniq stuff)
\$09	Front Desk Bus
\$0A	SANE (Fancy Floating Point stuff)
\$0B	Integer Math
\$0C	Text Tools
\$0D	<<<I don't know>>>
\$0E-\$20	Various RAM-based tools

The first eight tools in every tools set are all the same, and do not seem to be too important to the casual user. They include boot initialization code, version and status information, reset, and so on. Even a couple of spares.

Of interest to this article, tool number \$2A in set \$0B will convert a two-byte value to a four-character ASCII string. For example, \$12AF would be converted to the four bytes \$31, \$32, \$41, \$46. These are the ASCII values for the four hexadecimal digits of \$12AF. Lines 1090-1170 in the following program use this tool. Note that all inputs and outputs for the tool are handled through the stack.

Tool \$20 of tool set \$0C will print out a string in ASCIIZ format. ASCIIZ is a new term to me, which I discovered this week in the Microsoft book "Advanced MS-DOS". It means a string of ASCII characters terminated by a 00 byte. After lines 1140-1170 have placed the ASCII form of the number we converted into the four bytes at HEX (line 1290), we have an ASCIIZ string starting at MSG (lines 1280-1300). Pushing the



NEW !!!][IN A MAC: \$69.00

This Apple II emulator runs DOS 3.3 and PRODOS programs (including 6502 machine language routines) on a 512K Macintosh. All Apple II features are supported such as HI-RES/LO-RES graphics, 40/80 column text screens, language card and joystick. Also included: clock, RAM disk, keyboard buffer, on-screen HELP, access to the desk accessories and support for 4 logical disk drives. Package includes 2 MAC diskettes (PROGRAM holds emulation, communications and utility software, DATA holds DOS 3.3 and PRODOS system masters, including Applesoft and Integer BASIC) and 1 Apple II diskette (transfer software moves disk images to the MAC).

NEW !!! SCREEN.GEN: \$35.00

Develop HI-RES screens for your Apple II on a Macintosh. Don't be limited by MousePaint or other screen editors. Use MACPAINT (or any other application) on the MAC to create your Apple II screen. Then use SCREEN.GEN to transfer directly from the MAC to the Apple II (with SuperSerial card or equivalent). Package includes Apple II diskette with transfer software plus fully commented SOURCE code.

NEW !!! MIDI-MAGIC for Apple //c: \$49.00

Compatible with any MIDI equipped music keyboard, synthesizer, organ or piano. Package includes a MIDI-out cable (plugs directly into modem port - no modifications required!) and 6-song demo diskette. Large selection of digitized GRS player-piano music available for 19.00 per diskette (write for catalog). MIDI-MAGIC compatible with Apple II family using Passport MIDI card (or our own input/output card w/drum sync for only \$99.00).

FONT DOWNLOADER & EDITOR: \$39.00

Turn your printer into a custom typesetter. Downloaded characters remain active while printer is powered. Use with any Word Processor program capable of sending ESC and control codes to printer. Switch back and forth easily between standard and custom fonts. Special printer functions (like expanded, compressed etc.) supported. HIRES screen editor lets you create your own characters and special graphics symbols. For Apple II, II+, //e. Specify printer: Apple Dot Matrix, C.Itoh 8510A (Prowriter), Epson FX 80/100, or OkiData 92/93.

* The Font Downloader & Editor for the Apple Imagewriter Printer. For use with Apple II, II+, //e (with SuperSerial card) and the Apple //c (with builtin serial interface).

• FONT LIBRARY DISKETTE #1: \$19.00 contains lots of user-contributed fonts for all printers supported by the Font Downloader & Editor. Specify printer with order.

DISASM 2.2e : \$30.00 (\$50.00 with SOURCE Code)

Use this intelligent disassembler to investigate the inner workings of Apple II machine language programs. DISASM converts machine code into meaningful, symbolic source compatible with S-C, LISA, ToolKit and other assemblers. Handles data tables, displaced object code & even provides label substitution. Address-based triple cross reference generator included. DISASM is an invaluable machine language learning aid to both novice & expert alike. Don Lancaster says DISASM is "absolutely essential" in his ASSEMBLY COOKBOOK.

The 'PERFORMER' CARD: \$39.00 (\$59.00 with SOURCE Code)

Converts a 'dumb' parallel printer I/F card into a 'smart' one. Command menu eliminates need to remember complicated ESC codes. Features include perforation skip, auto page numbering with date & title. Includes large HIRES graphics & text screen dumps. Specify printer: MX-80 with Graftrax-80, MX-100, MX-80/100 with Graftraxplus, NEC 8092A, C.Itoh 8510 (Prowriter), OkiData 82A/83A with Okigraph & OkiData 92/93.

'MIRROR' ROM: \$25.00 (\$45.00 with SOURCE Code)

Communications ROM plugs directly into Novation's Apple-Cat Modem card. Basic modes: Dumb Terminal, Remote Console & Programmable Modem. Features include: selectable pulse or tone dialing, true dialtone detection, audible ring detect, ring-back, printer buffer, 80 col card & shift key mod support.

RAM/ROM DEVELOPMENT BOARD: \$30.00

Plugs into any Apple slot. Holds one user-supplied 2Kx8 memory chip (6116 type RAM for program development or 2716 EPROM to keep your favorite routines on-line). Maps into \$Cn00-CnFF and \$C800-CFFF.

C-PRINT For The APPLE //c: \$69.00

Connect standard parallel printers to an Apple //c. C-PRINT plugs into the standard Apple //c printer serial port and into any printer having a standard 36 pin centronics-type parallel connector. Just plug in and print!

Unless otherwise specified, all Apple II diskettes are standard (not copy protected!) 3.3 DOS.

Avoid a \$3.00 handling charge by enclosing full payment with order.

VISA/MC and COD phone orders OK.

RAK-WARE 41 Ralph Road W. Orange N J 07052 (201) 325-1885



address of MSG onto the stack and calling for tool \$200C will print out the string on the screen. See lines 1180-1220.

Line 1190 may need a word of explanation. The tool needs a four-byte address on the stack, so line 1190 pushes the high-order two bytes of the address. I could have used "PEA MSG/65536", but I like "PEA MSG/256/256" better.

If you have a //gs, try out this little example. It will get you started in understanding all the new tools you have in your toolbox. If you don't have a //gs yet, start saving your nickels!

```

1000 *SAVE S.PRINT.DEMO
1010 *-----
1020 .OP 65816
1030 T
000800- 18 1040 clc GO INTO NATIVE MODE
000801- FB 1050 xce
000802- C2 30 1060 rep ##30 FULL 16
000804- A9 34 12 1070 lda ##$1234 SAMPLE VALUE IN A-REG
1080 *---Tool: convert hex to string-----
000807- F4 00 00 1090 PEA 0 4-BYTES FOR RESULT
00080A- F4 00 00 1100 PEA 0
00080D- 48 1110 PHA VALUE TO BE CONVERTED
00080E- A2 0B 2A 1120 LDX ##$2A0B TOOL: HEX-VALUE TO ASCII
000811- 22 00 00 E1 1130 JSL $E10000
000815- 68 1140 PLA TWO CHARS
000816- 8D 4E 08 1150 STA HEX INTO STRING
000819- 68 1160 PLA TWO MORE CHARS
00081A- 8D 50 08 1170 STA HEX+2 INTO STRING
1180 *---Tool: print a string-----
00081D- F4 00 00 1190 PEA MSG/256/256
000820- F4 2D 08 1200 PEA MSG ADDRESS OF STRING
000823- A2 0C 20 1210 LDX ##$200C TOOL: PRINT STRING
000826- 22 00 00 E1 1220 JSL $E10000
1230 *-----
00082A- 38 1240 SEC RETURN IN EMULATION MODE
00082B- FB 1250 XCE
00082C- 60 1260 RTS
1270 *-----
00082D- 54 48 45 20
000831- 4E 55 4D 42
000835- 45 52 20 49
000839- 4E 20 54 48
00083D- 45 20 41 2D
000841- 52 45 47 49
000845- 53 54 45 52
000849- 20 49 53 20
00084D- 24 1280 MSG .AS "THE NUMBER IN THE A-REGISTER IS $"
00084E- 1290 HEX .BS 4
000852- 0D 00 1300 .HS 0D.00
1310 *-----

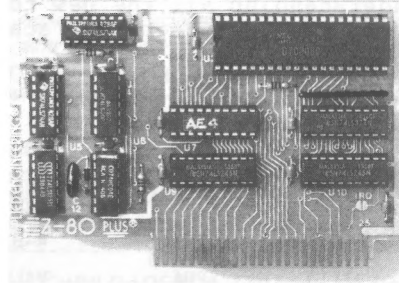
```

Woz Re-Codes Hi-Res Address Calculations....Bob Sander-Cederlof

In the October or November issue of the Washington Apple Pi newsletter, Rick Chapman wrote a review of various methods of calculating the hi-res base addresses. Steve Wozniak liked the article, and responded with a long "letter to the editor" in the December issue. Steve also presented a new version of the hi-res address calculator which is both shorter and faster. In fact, as far as I am aware, it is the fastest method ever, except for table-lookups.

In the September 1983 issue of Apple Assembly Line, I presented both the original Woz code and a shorter-faster version by Harry Cheung of Nigeria. Here are the specs:

With the Z-80 Plus®, run over 5000 new CP/M® programs.



*Now, get two computers
in one, and all the
advantages of both.*

Enter the CP/M world with the new Z-80 Plus card from Applied Engineering, and introduce your Apple IIe, IIgs or II+ to thousands of CP/M programs. Only the Z-80 Plus comes standard with the new 5.1 software, the most advanced system ever for running CP/M programs.

The new 5.1 boasts advanced features like built-in disk emulation for popular memory expansion boards, boosting both system speed and storage capacity. And menu-driven utilities that let you get to work faster. The Z-80 Plus also lets you run older CP/M programs—all the way down to Version 1.6 (2.2 is the most popular).

The Z-80 Plus is the only card on the market capable of accessing more than 64K. If you have an extended 80-column card, all 128K is usable. And if you have RamWorks, RamFactor, sRam, sRam Plus, or an Apple memory card in your IIe, IIgs or II+, up to 1088K is available.

Each Z-80 Plus comes with our CP/M Ram Drive software, enabling most memory cards to work as a high-speed Ram disk so CP/M software can run up to *twenty times faster*. So packages like WordStar, dBASE II, Turbo Pascal and most others run at blinding speed.

Simply plug the Z-80 Plus into any slot in your Apple. You'll get the benefits of two computers in one—all at an unbelievably low price (only \$149!).

- Fully compatible with ALL CP/M software
- Fully compatible with 5¼" and 3½" disk drives as well as most hard disks including the Sider
- Fully compatible with Microsoft disks (no pre-boot required)
- Specifically designed for high speed operation
- Runs WordStar, dBASE II, Turbo Pascal, and ALL other CP/M software with *no pre-boot*
- Semi-custom I.C. and low parts count allows Z-80 Plus to fly through CP/M programs with extremely low power consumption (we use the Z-80B)
- Does EVERYTHING other Z-80 boards do, *plus* Z-80 interrupts
- Five-year warranty

To get your Z-80 Plus or for more information see your dealer, or call 214-241-6060 9 a.m. to 11 p.m. seven days, or send check or money order to Applied Engineering, MasterCard, VISA and C.O.D. welcome. Texas residents add 5½% sales tax. Add \$10.00 outside U.S.A.

AE Applied Engineering™
P.O. Box 798, Carrollton, TX 75006
(214) 241-6060

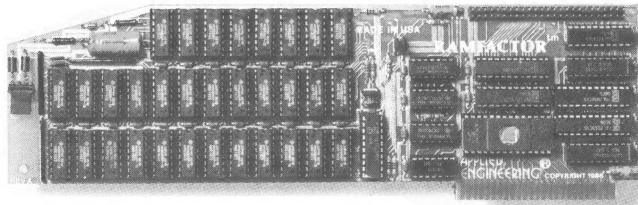
CP/M is a registered trademark of Digital Research.

RamFactor™

All the Performance, Speed, and Software Compatibility of RamWorks™ in a Slot 1 through 7 Card.

That's right! Now Applied Engineering offers you a choice. While RamWorks is the clear winner for the auxiliary slot in a IIe, RamFactor is the standard for slots 1 through 7. Now anyone with an Apple II+, Franklin, or Apple IIe preferring to use slots 1 through 7 can now enjoy the speed and performance that until now was only available with RamWorks.

With RamFactor, you'll be able to instantly add another 256K, 512K, or a full 1 meg on the main board and up to 16 meg with additional piggyback card. And since virtually all software is automatically compatible with RamFactor, you'll immediately be able to load programs into RamFactor for instantaneous access to information. You'll also be able to store more data for larger word processing documents, bigger data bases, and expanded spreadsheets.



Very Compatible

All the leading software is already compatible with RamFactor. Programs like AppleWorks, Pinpoint, BPI, Managing Your Money, Dollars and Sense, SuperCalc 3A, PFS, MouseWrite, MouseDesk, MouseCalc, Sensible Speller, Applewriter IIe, Business Works, ReportWorks, Catalyst 3.0 and more. And RamFactor is fully ProDos, DOS 3.3, Pascal 1.3 and CP/M compatible. In fact, no other memory card (RamWorks excepted) is more compatible with commercial software.

AppleWorks Power

There are other slot 1-7 cards that give AppleWorks a larger desktop, but that's the end of their story. But RamFactor is the only slot 1-7 card that increases AppleWorks internal memory limits, increasing the maximum number of records in the database and lines permitted in the word processor, and RamFactor is the only standard slot card that will automatically load AppleWorks into RAM dramatically increasing speed and eliminating the time required to access the program disk, it will even display the time and date on the AppleWorks screen with any ProDos clock. RamFactor will automatically segment large files so they can be saved on 5¼", 3½", and hard disks. All this performance is available to anyone with an Apple IIe or II+ with an 80

column card. RamFactor, no other standard slot card comes close to enhancing AppleWorks so much.

True 65C816 16 Bit Power

RamFactor has a built-in 65C816 CPU port for direct connection to our IIe 65C816 card for linearly addressing up to 16 meg for the most powerful 16 bit applications. (II+ 65C816 card under development.)

Powerful Program Switcher

With RamFactor, you can organize memory into multiple work areas and switch between them. Each work area can contain different programs and even different operating systems. Now you can switch from one program to another or even switch from AppleWorks to DOS 3.3 to CP/M to Pascal to ProDos in under a second. And with our Battery back-up option, you can have permanent storage for up to 20 years.

Quality and Support of the Industry Leader

RamFactor is from Applied Engineering, the largest, most well supported manufacturer of Apple peripherals and the inventor of large RAM cards for the Apple. With our 5 year no hassle warranty and outstanding technical support, you're assured of the most trouble free product you can buy.

Features:

- Up to 16 meg total memory, 256K to 1 meg on main board. Up to 16 meg with additional memory on piggyback card.
- Fully Apple II Memory Expansion compatible
- Compatible with Apple IIe, II+ and Franklin
- Battery back-up option allows you to turn on your Apple and run your favorite programs in less than 1 second!
- Automatically recognized by ProDos, DOS 3.3, Pascal and CP/M
- Built-in RamDrive™ software (a true RAM disk not disk caching)
- Systems are directly bootable from RamFactor if desired
- Built-in linear addressing 16 bit co-processor port
- Built-in self diagnostic software
- Automatic expansion with AppleWorks 1.3 or later
- Allows Apple II+ and IIe to run your AppleWorks without buying additional software
- Accelerates AppleWorks
- Displays time and date on the AppleWorks screen with any ProDos clock
- Fits any I/O slot except slot 3
- Fully socketed and user upgradeable
- Much, much more

RamFactor with 256K	\$239
RamFactor with 512K	\$289
RamFactor with 1 MEG	\$389
RamFactor with 2-16 MEG	CALL
Battery Back-up Option	\$179
65C816 16 Bit Card	\$159

Order RamFactor today... with 15 day money back guarantee and our "no hassle" five year warranty. Call 9 a.m. to 11 p.m., 7 days, or send check or money order to Applied Engineering, MasterCard, Visa and C.O.D. welcome. Texas residents add 5½% sales tax. Add \$10.00 if outside USA.

AE Applied Engineering™
The Apple enhancement experts.
(214) 241-6060

P.O. Box 798, Carrollton, TX 75006

Applesoft ROM version: 33 bytes, 61 cycles
 Harry Cheung version: 25 bytes, 46 cycles
 New Wozniak version: 26 bytes, 36 or 37 cycles

The byte counts do not include an RTS at the end of the code, nor do the times include a JSR-RTS. After all, if you are really working for speed you will put the code in its place, not make it a subroutine.

Woz's new version takes either 36 or 37 cycles, depending on the values for the first two bits of the line number. Remember that the line number can be any value from 0 to 191, or \$00...\$BF. That means the first two bits are either 00, 01, or 10. If you look at lines 1090-1120 below, you will see that the shortest path is for 00, taking both branches, giving a running time for the whole calculation of 36 cycles. If the first two bits are 01 or 10, one branch will be taken and the other not, making the total time 37 cycles. In Woz's letter he shortchanged himself, thinking possibly both branches might not be taken, giving a total running time of 38 cycles; this cannot happen with legal line numbers.

Line 1180 adds in either \$10 or \$20, depending on which hi-res page you are using. The Applesoft code here adds in a value of either \$20 or \$40, so if this version were to be inserted into Applesoft the generation of HPAG2 would have to be changed. No problem, and not likely anyway. By the way, if you are only using one specific hi-res page, you can change line 1180 to an immediate mode form, saving yet another cycle.

Here is Woz's new version, reformatted for the S-C Assembler and with some changes in comments:

```

1000 *SAVE NEW.WOZ.HIRES.CALC
1010 *-----
16- 1020 GBASL .EQ $26
17- 1030 GBASH .EQ $27
E6- 1040 HPAG2 .EQ $E6           Applesoft puts it here anyway.
1050 *-----
C800- 0A 1060 CALC ASL A--BCDEFGH0
C801- AA 1070 TAX TAX...TXA could be TAY...TYA
C802- 29 F0 1080 AND #$F0 A--BCDE0000
C804- 10 02 1090 BPL .1 B=0
C806- 09 05 1100 ORA #$05 A--BCDE0BOB
C808- 90 02 1110 .1 BCC .2 A=0
C80A- 09 0A 1120 ORA #$0A A--BCDEABAB
C80C- 0A 1130 .2 ASL B--CDEABAB0
C80D- 0A 1140 ASL C--DEABAB00
C80E- 85 26 1150 STA GBASL C--BCDEFGH0
C810- 8A 1160 TXA C--0000FGH0
C811- 29 0E 1170 AND #$0E C--00xxFGHC
C813- 65 E6 1180 ADC HPAG2 0--00xxFGHC
1190 * HPAG2 = $10 for base $2000, $20 for base $4000
0815- 06 26 1200 ASL GBASL D--00xxFGHC GBASL=EABAB000
0817- 2A 1210 ROL 0--0xxFGHCD
0818- 85 27 1220 STA GBASH
081A- 60 1230 RTS
1240 *-----

```

Apple Assembly Line (ISSN 0889-4302) is published monthly by S-C SOFTWARE CORPORATION, P.O. Box 280300, Dallas, Texas 75228. Phone (214) 324-2050. Subscription rate is \$18 per year in the USA, sent Bulk Mail; add \$3 for First Class postage in USA, Canada, and Mexico; add \$14 postage for other countries. Back issues are available for \$1.80 each (other countries add \$1 per back issue for postage). A subscription to the newsletter and the Monthly Disk containing all source code is \$64 per year in the US, Canada and Mexico, and \$87 to other countries.

All material herein is copyrighted by S-C SOFTWARE CORPORATION, all rights reserved. (Apple is a registered trademark of Apple Computer, Inc.)